Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC 2020)
IEEE Xplore Part Number: CFP20J06-ART; ISBN: 978-1-7281-2813-9

PID based locomotion of multi-terrain robot using ROS platform

Rajesh Kannan Megalingam, Deepak Nagalla, Katta Nigam, Vamsi Gontu, Phanindra Kumar Allada,
Department of Electronics and Communication Engineering,
Amrita Vishwa Vidyapeetham, Amritapuri, India
rajeshkannan@ieee.org, nagalla.deepak@gmail.com, nigamkatta@gmail.com, vamsi.gontu2308@gmail.com, alladaphanindrakumar@gmail.com

***Abstract*—This work deals with the implementation of closed-loop control on the multi-terrain robots. The robots which are working in these terrains should be robust and are easily prone to deviate from the expected path due to the rugged path. An open-loop control on these robots will not correct this misdemeanor. An efficient method to fuse wheel encoder and Inertial measuring unit (IMU) data for Proportional - Integral - Derivative (PID) control of the wheeled robot in uneven pitches was proposed. The method to fuse this data available from different nodes using the Robot Operating System (ROS) was also described. The algorithms and methods proposed in this work can be implemented for odometry data used by Simultaneous Localisation and Mapping (SLAM) packages in ROS.**

Keywords*—PID, Tuning, Control Systems, ROS, IMU, Encoder, Uneven terrain, Stability, Human Machine Interface (HMI)

I. INTRODUCTION

Robotics has been extending its wings to almost all the varied fields. It has become so influential that RPA is one of the most valued and predominant in any industry be it an IT MNC or a deep tech startup. By that token, implementation of wheeled robots or tracked robots in uneven terrains for military, rescue, exploration and many other motives has been more prevalent in the past few years. But, the movement of the robot on these rugged terrains might not be as expected. It mostly deviates from the expected path, when operated with open-loop control like any other even terrain robot. Closed-Loop control is required especially when the deviation is too large that the expected path can not be retracted. When the location of the robot should be tracked, as in SLAM applications, data from a single sensor might not be sufficient in most of the terrains as there always will me miss alignment in data received and original data or position of the robot. So, fused data from different sensors is needed to implement closed-loop control efficiently. Considering the case that different sensors might be connected to different microcontroller units (MCU's), a ROS based technique for sending the data to the master controller on the locomotive and fusing the data in a central node receiving information from different MCU's is described in detail in this paper.

II. MOTIVATION AND PROBLEM STATEMENT

Several robots malfunction while undergoing a series of tests. One of the common malfunctions is the deviation from its intended path. Though theoretically, every system is considered to be identical there might some misalignments in its movements that cause the robot to deviate from its original path. There could be scenarios where the robot might not follow the specified path. There can be situations where the control mechanism is simple but the robot behavior can be complicated. Faulty values or the difference in the input signal from the system might lead to the error in the movement and sometimes if the behavior is unexpected it might lead to damage to the robot. So, implementing an efficient method for the control of the wheeled robot in uneven terrains is preferred.

There are several methods to resolve this error like PID control of a robot, which can be used to control the speed and other process variables. PID controllers use a closed-loop feedback mechanism to control process variables and gives the most accurate readings. Faulty values or the difference in the input signal from the system might lead to this problem which can be resolved by using data from different sensors to adjust the misalignments and move the robot in a correct path.

III. RELATED WORKS

In paper [1], the authors discussed various domains required for a rescue robot. The authors also proposed solutions for the stable motion of rescue robots on uneven terrains. As we know, design and mechanism play a major role in the stable locomotion of the robot. Paper [2] discusses the design and testing of a tracked robot on several uneven terrains. Every design will have drawbacks in the moment of the robot in some terrains. Due to the instability in some terrains, the robot loses its path or heading. To overcome the error Proportional, integral and derivative control can be used by binding encoder sensor data. Paper [3] discusses how to correct speed and heading using encoder data. The authors used a closed-loop PID control algorithm to get speed control of the robot accurately. PID values need to be tuned perfectly for accurate outcomes. In paper [4] the authors discuss the tuning methods for accurate PID values based on the need and specification. The paper [5] discusses a self-balancing two-wheeled robot using Qt-Creator with the use of effective controllers such as PID. Paper [6] discusses the PID control of the two-wheeled robot. The authors used a gyroscope and accelerometer for the stable motion of the two-wheeled robot using PID. In paper[7], the authors discuss that the tuned values may lead to problems in some cases. They were proposing an intelligent learning algorithm that updates the PID values. In paper [8], the authors discuss the self-balancing algorithm based on PID using IMU gyroscope and accelerometer values for the stable motion of the drone. Paper [9] discusses the recent design and tuning methods of PID controllers. It also describes process model identification, PID controller design, etc. Paper [10] discusses a robust scan matching approach using the LIDAR and 3D attitude estimation system for grid mapping based on inertial sensing. It can be implemented in various challenging environments for mapping and localization using ROS. Paper [11] discusses controlling a robot in different simulated environments and movement of the robot with different speeds in multiple scenarios. Paper [12] discusses the PID control scheme for a twin-rotor motor for the application of 2-DOF motion using a non-linear system to attain a particular position and follow the trajectory.

IV. ARCHITECTURAL DIAGRAM

Fig 2 System Architecture

Fig 2 delineates the overall working of the system integrated with the ROS platform. HMI, which is connected to the master controller at the control station on which ROS_Master is running, will be used to send the control instructions to the robot and visualize the camera feed and other sensory feedback from the robot. Wireless communication between the ROS master node at the control station and ROS master node running on the robot can be established via the WiFi interface. We can make necessary adjustments in the bashrc file of the Linux system on which the ROS is running, to make the system on the robot as a slave to the system at the control station so as to entitle the operator with complete command over the slave system. ROS master is the central node of the entire ROS system, which should be launched prior to running any other functions pertaining to ROS.

A. HMI Node:

HMI used in this system is a single two-axis joystick, which can be used to send velocity values to the robot which enables the operator to vary the speed at which the robot is driven. The HMI node establishes the connection between the real-time values from the joystick, available with the rosserial node of the control station and the uC_node of the robot. The commands are published over a topic named /cmd_vel, which connects the two nodes mentioned above. ROSSerial node is used to establish serial communication between the microcontroller and the computer for the exchange of data.

B. PID_Node:

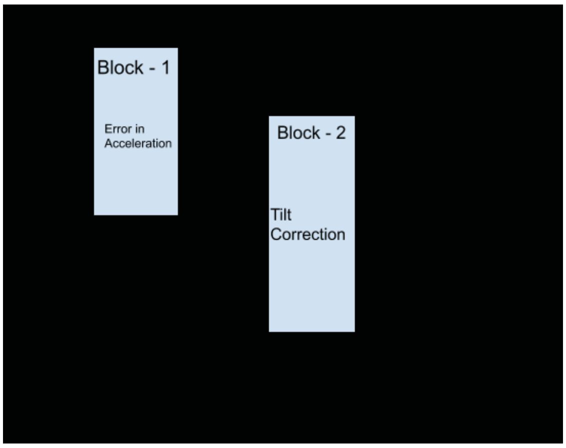

Fig 3 PID NODE Architecture

Fig 3 depicts how the proposed PID control method works and can be implemented in a single node of ROS, called PID_Node (in this work). The control instructions or commands from HMI and the current acceleration value are collected at Block - 1, which calculates the error between the current acceleration and the desired acceleration to give the error in acceleration. Block - 2 takes in encoder data to calculate the deviation or tilt from the actual path, which will be fine-tuned by the additional tilt value calculated from IMU data. The P, I, D values can be given by the user based on the requirement and the desired acceleration for each motor is calculated.

C. u C_Node:

This node reflects the working of an MCU, which takes in sensory data from all the available sensors and gives signaling for the control of motors to the motor drivers. This node sends the encoder and IMU data to the PID_Node and accepts acceleration values and generates respective signals to drive the motors. This node in this work can be launched only by first starting a rosserial node.

D. ROS_Master:

This master node should be running on each of the master controllers involved in performing the task. This acts as a key to the establishment of communication and exchange of information between different nodes running on the same or different master controllers.

V. DESIGN AND IMPLEMENTATION

In this paper, robot design plays an important role in the implementation of PID control over the multi-terrain paths. A tracked wheel robot that can navigate in any type of clustered areas is designed to overcome a few problems that can be faced during the implementation of the PID control. Closed-loop control is included in the system in such a way that it can get the feedback and to rectify the error which can have entire control within the system. The below figure shows and explains the background work in the robot.

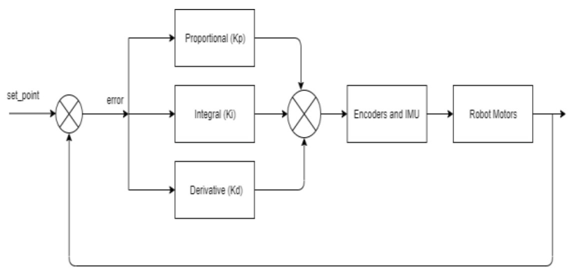

Fig 1 Representation of closed Loop system of PID

In Fig 1, initial value(input) is given as a setpoint from the operator through HMI. Gain constants (P, I, D) mentioned in the fig(1) are used in error correction. Error correction can be achieved by performing multiple tunings. Proportional gain helps to reduce the rise time simultaneously there is a sudden increase in the overshoot which leads to instability of the system. Similarly integral gain may lead to lessen in the steady-state error but increase in oscillations, differential gain put forwards the increment in the steady-state errors but at the same time reducing the overshoots. In tracking and correcting the error by applying different tuning processes such as P, PI, PD, PID provides effective results. All the above processes come under self-tuning methods, but there are other types of tuning methods which include Ziegler Nichols method, linear matrix inequalities and convex optimization methods give different approaches to detecting and correcting errors. The main concern of this Zigler Nichols is to find the gain stability of the system at a given specific oscillation frequency. The below equation shows the basic representation of backend calculations of PID to track the error in the system.

$$correction = P.e(t) + \int I.e(t)dt \ + D.\ \frac{de(t)}{dt}$$

By using a differential drive motor control with integrated IMU and encoders the robot can navigate the path. When the initial setpoint is given, the internal robot performs all the calculations using the gain constants and sends those values to encoders and then to motors of the robot. This leads to an abrupt impede in speed in one of the motors and diminished in the other. To get over this problem, implementation of PD tuning gives the position of the robot and implementation of PI will give the tilt or the inclination angle of the robot with the ground. To attain complete stability of the robot weight, height includes the major aspects of the design which can decrease the gain constant values leads to reduce in the error.

For implementing the above PID algorithm, the ROS platform is used as an interface between the gazebo simulator and the mobile robot. Gazebo helps to evaluate the above algorithm by implementing in different terrains. The algorithms for calculating the deviation of the robot's path can be clearly seen in plain surfaces that already exists. But for implementations of the PID algorithm in different terrains in a gazebo can give accurate measures and if any troubleshooting occurs it can be known immediately after launching of files.

One of the main parts of the implementation will depend on the hardware interface. While considering hardware interface it includes the design of the body, electrical circuit, HMI. The design of the robot plays a vital role in the alignment of the path. Even if there is a slight error in the design of the robot or if the driven current from the system is different from the intended value, the robot does not follow the correct path. While this does not happen in the case of simulation as there cannot be factors such as an error in driving current or design.

VI. EXPERIMENTS AND RESULTS

The main goal of designing a multi-terrain robot is to give a robust motion in all types of terrains. In most cases, the robot structure and uneven terrains can lead to error in its motion. So, in order to solve the error, there should be a control algorithm that simultaneously adapts itself to error and corrects itself. So, using a control application like a PID controller can mitigate the error caused due to the uneven terrains. And the main motive to use the PID algorithm is to correct the direction heading without any devious motion. Pid is passed with fused data of IMU and encoder as an input for a setpoint reference for the controller. These data are used for error checking and correction in the motion of the robot on irregular terrains. We have tested our multi-terrain robot on uneven terrain with and without PID implementation.

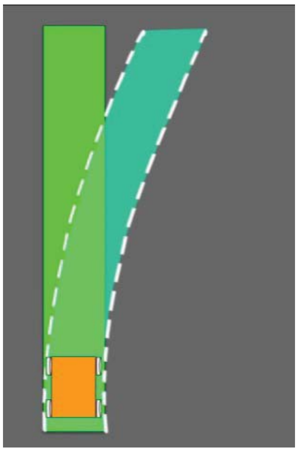

Fig 4. Depiction of the path followed by the robot without PID control

Fig 4 depicts is a clear picture of how the robot without PID deviates from the desired path. Tuning the PID values can change the behavior of the motion. So, we tuned Proportional, Integral and Derivative constants values by experimentation. On rigorous analysis of the outcomes and the behavior of the motion of the robot, we chose the P = 0.6, I = 0.3, D = 0.5 for our robot. Fig 5 depicts how the PID implemented robot tries to maintain a straight path.

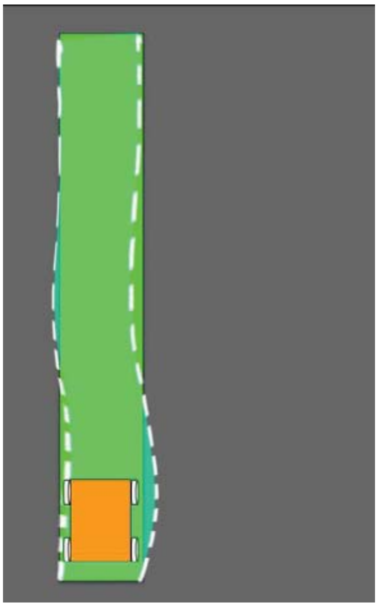

Fig 5. Depiction of the path followed by the robot after implementation of the PID algorithm

From that observation, we plotted the encoder values of the left and right motor. The graph in Fig 5 depicts the encoder readings without PID implementation. The graph in Fig 6 depicts the encoder readings with PID implementation. The blue line represents the readings of the left tracked wheel encoder and the red line indicates the reading of right tracked wheel encoder.

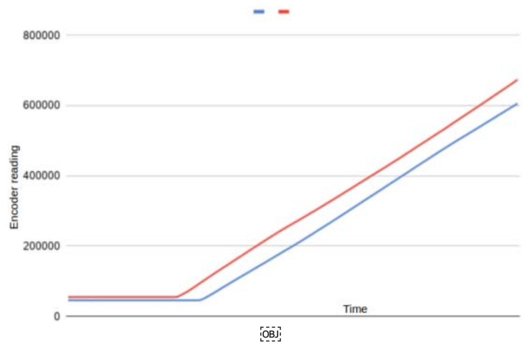

Fig. 5. Encoder readings without PID implementation

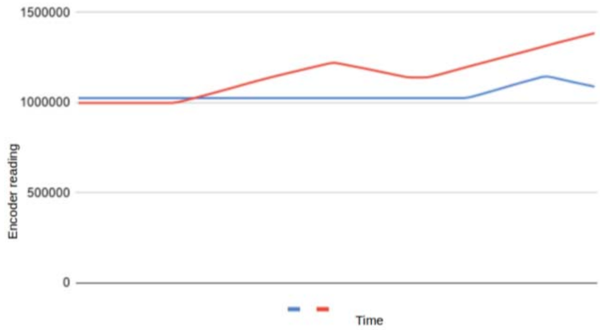

Fig.6. Encoder readings with PID implementation

From the two graphs of encoder readings, we can observe that the robot tried to align its orientation in a straight manner by adjusting itself. That's the reason why there is a fluctuation in the encoder readings when we compare the behavior with and without the PID controller.

VII. FUTURE WORKS

In the further revision of the work, the light will be thrown on the implementation of the proposed algorithm in SLAM and also evaluate different SLAM packages available on the ROS platform with the fused data obtained from encoders and IMU. As the robot is robust, there is a chance of encoders to be damaged due to jerks if the mechanical design has glitches. So, the further revision will include encoder less closed-loop PID control, similar to the efficient encoder less LIDAR and IMU-based localization techniques explained in [10].

VIII. CONCLUSION

A successful solution for eradicating path deviation in multi-terrain robots was proposed and implemented. The deviation is almost negligible after the implementation of the proposed algorithm. But, when implemented on heavy mobile platforms without proper suspension or platforms where the mounting of encoders is not precise, it was observed that the encoders are being affected by the jerks and rugged movement. So, the mechanical aspects should be taken care of, for the proposed algorithm to work efficiently. Though the closed-loop control is implemented, it was observed that there is a minute deviation from the original path when the path is too rugged. This will not affect the mission which involves complete teleoperation of the robot as the path can be retracted easily due to minute deviation.

IX. ACKNOWLEDGMENT

We are gratified to express our gratitude towards the Electronics and Communication Engineering Department and HuT Labs of Amritapuri campus of Amrita Vishwa Vidyapeetham University for their ceaseless succor, without which this work would not have been progressed.

REFERENCES

[1] Yi Liu, Yuhua Zhong, Xieyuanli Chen, Pan Wang, Huimin Lu, Junhao Xiao, Hui Zhang, "The design of a fully autonomous robot system for urban search and rescue," 2016 IEEE International Conference on Information and Automation (ICIA), Ningbo, 2016, pp. 1206-1211.

[2] Zhang Guowei, Li Bin, Li Zhiqiang, Wang Cong, Zhang Handuo, Shang Hong, Hu Weijian, Zhang Tao "Development of robotic spreader for earthquake rescue," 2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014), Hokkaido, 2014, pp. 1-5.

[3] Jie Meng, Anbang Liu, Yiqin Yang, Zhe Wu, and Qingyang Xu, "Two-Wheeled Robot Platform Based on PID Control," 2018 5th International Conference on Information Science and Control Engineering (ICISCE), Zhengzhou, 2018, pp. 1011-1014.

[4] P. Cominos and N. Munro, "PID controllers: recent tuning methods and design to specification," in IEE Proceedings - Control Theory and Applications, vol. 149, no. 1, pp. 46-53, Jan. 2002.

[5] Ali Unluturk, Omer Aydogdu and Ufuk Guner, "Design and PID control of two wheeled autonomous balance robot," 2013 International Conference on Electronics, Computer and Computation (ICECCO), Ankara, 2013, pp. 260-264.

[6] Derry Pratama, Fernando Ardilla, Eko Henfri Binugroho and Dadet Pramadihanto, "Tilt set-point correction system for balancing robot using PID controller," 2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, 2015, pp. 129-135.

[7] Li-xin Ma, Dao-nian Shi, Min-xuan Chen and Xiao-qin Wang, "Application of intelligent PID control for robot," 2008 IEEE Conference on Cybernetics and Intelligent Systems, Chengdu, 2008, pp. 455-458.

[8] R. K. Megalingam, R. V. R. Raj, A. Masetti, T. Akhil, G. N. Chowdary and V. S. Naick, "Design and Implementation of an Arena for Testing and Evaluating Quadcopter," 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1-7.

[9] Hang Wu, Weihua Su, Zhiguo Liu, "PID controllers: Design and tuning methods," 2014 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, 2014, pp. 808-813.

[10] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in (IEEE) International Symposium on Security, Safety, and Rescue Robotics, in September 2011 at Kyoto, Japan.

[11] Rajesh Kannan Megalingam, Deepak Nagalla, Vamsi Gontu, Ravi Kiran Pasumarthi, Allada Phanindra Kumar, “ ROS Based, Simulation and Control of Wheeled Robot using Gamer’s Steering Wheel ”, International Conference on Computing Communication and Automation (ICCCA), Jul. 2019.

[12] P. J. Neeraj and P. N. Seema, "Control of Twin Rotor MIMO System Using Cross PID Control Technique," *2018 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0744-0750, Nov. 2018.