AN METHOD TO STOP DIFFERENTIAL MODELS OF FALLING OVER

Tiago Barretto Sant'Anna[1], Rebeca Tourinho Lima[1]

[1] *Robotics dept. SENAI CIMATEC, Brazil*

Abstract: Robotics is a rapidly growing field with a profound impact on various industries, aiming to design machines that assist humans in practical tasks. Autonomous navigation, a critical aspect of robotics, enables robots to move and operate independently without human intervention. However, navigating different environments presents numerous challenges, including the risk of robots tipping over, leading to damage or potential harm to humans. This article proposes a solution to prevent tipping in differential robots during autonomous navigation. The solution employs behavior trees, a computationally efficient approach, integrated into the robot's navigation system. By continuously monitoring data from an IMU sensor, the plugin effectively detects oscillatory movements and takes corrective actions, ensuring stable and safe robot operation.

Keywords: robotics; autonomous; differential robots; falling over.

UM MÉTODO PARA EVITAR QUE MODELOS DIFERENCIAIS TOMBEM.

Resumo: A robótica é uma área em rápida expansão com impactos profundos em várias indústrias, tendo como o objetivo de projetar máquinas que auxiliam os seres humanos em tarefas práticas. A navegação autônoma, um aspecto crítico da robótica, permite que os robôs se movam e operem independentemente, sem intervenção humana. No entanto, a navegação em ambientes diferentes apresenta inúmeros desafios, incluindo o risco de que os robôs possam tombar, causando danos ou potencialmente prejudicando os seres humanos. Este artigo propõe uma solução para evitar o tombamento de robôs diferenciais durante a navegação autônoma. A solução utiliza árvores de comportamento, uma abordagem computacionalmente eficiente, integrada ao sistema de navegação do robô. Ao monitorar continuamente os dados de um sensor IMU, o plugin detecta efetivamente movimentos oscilatórios e toma ações corretivas, garantindo uma operação estável e segura do robô.

Palavras-chave: Robótica; autônomo; robôs diferenciais; tombar.

1. INTRODUCTION

Robotics is a rapidly growing field that involves the design, construction, operation, and use of robots [1]. The goal of robotics is to design machines that can help and assist humans in various practical purposes, whether domestically, commercially, or militarily [1]. Today, robots are changing our lives, making our daily life easier and assisting in producing more products [2][3].

Autonomous navigation is a crucial aspect of robotics enabling robots to move and operate independently without human intervention [4]. Autonomous navigation involves the use of various sensors and algorithms to enable robots to perceive their environment, make decisions, and move around obstacles [5]. The sensors used in autonomous navigation include LiDAR, which is used for scanning, localization, obstacle detection, and accurate depth perception [4]. Perception algorithms are used to continuously scan and track the surrounding environment through various types of available sensors, including radar, LiDAR, or cameras, to emulate human vision [4]. Localization and mapping are also important aspects of autonomous navigation, as they enable the robot to determine its location and navigate to its destination [4]. Autonomous navigation is a rapidly evolving field that has the potential to revolutionize various industries, including manufacturing, transportation, and healthcare [6].

To navigate is intrinsic to consider the environment a robot has to travel. Autonomous navigation in different environments demands several challenges that need to be addressed to ensure the safe and efficient operation of robots. Among the key challenges faced in this domain are the navigation in unfamiliar environments, where robots must function with limited prior knowledge or structure [7]. Negotiating highly-constrained spaces, such as crowded areas or narrow corridors, also presents difficulties, requiring the robots to detect and evade obstacles skillfully [8]. Furthermore, autonomous navigation in unstructured environments, particularly outdoors, demands unique hurdles as they lack clearly discernible paths and landmarks to assist the robot's movement [9]. Additionally, rough terrain requires sophisticated terrain analysis capabilities to determine navigability and plan appropriate paths [10]. Above all, safety remains a critical concern in autonomous navigation, necessitating robots to adeptly identify and avoid obstacles, including humans, to prevent accidents [5].

Different types of terrains can offer a risk of differential robots falling over themselves, which can lead to damage to the robot or even injury to humans. Mobile robots can encounter negative obstacles on their route, such as stairs, docks, ledges, and potholes [11]. The terrain inclination or slope can influence the orientation angles of the robot, which is important to consider in path planning for autonomous mobile robots [10]. Loose soils with a high content of stony elements, obstacles, or hollows can cause the robot to slip and fall [12]. In some cases, robots can collide with humans, leading to injuries. A study has shown that both adults and children are at risk of severe head injuries from falling to the ground after colliding with personal mobility devices and service robots [13]. To ensure operational safety, standards such as ISO 15066:2016 have been established for industrial robots [13].

In this way, this article aims to develop a solution to prevent robots from tipping over. A simple, computationally low-cost solution was sought, which could be easily integrated into the navigation system. To achieve this, a development was carried out using behavior trees, due to their consolidation in autonomous navigation. Thus, a behavior tree plugin was created to monitor the data sent by an IMU sensor, process it, and notify if there is a possibility of tipping over.

2. METHODOLOGY

The methodology for this project involved a well-structured and comprehensive approach, comprising several key stages that are outlined below. Each stage was carefully planned and executed to ensure the successful development of the solution:

2.1 Problem Understanding and Knowledge Acquisition

The first step in the development process was to gain a thorough understanding of the problem at hand. Extensive research was conducted, involving a comprehensive search for relevant articles and literature in the specific area. In our extensive search across Scopus and Web of Science databases, we uncovered approximately 14 articles related to the topic of robots and stability. However, upon closer examination, only five of these articles offered insights and solutions pertaining to the issue of robots falling over. Regrettably, none of these five articles specifically addressed the concerns related to differential robots. Through the analysis of these sources, valuable insights were gathered, allowing for a deeper comprehension of the physical phenomena associated with the desired behavior.

2.2 Software Development

To address the problem, a range of software tools were employed in the development process. The primary tool utilized was the robotics framework ROS2 [14], which offers a collection of software libraries and tools designed to facilitate the creation of robot applications. Additionally, the Navigation2 [15] package was employed to enable autonomous navigation, allowing the robot to autonomously reach designated goal states based on specific maps.

A crucial component of the solution involved the creation of a plugin using the BehaviorTreeCpp [16] library within ROS2. This library played a central role in designing and implementing state trees, essential for governing the robot's decision-making process.

3. RESULTS AND DISCUSSION

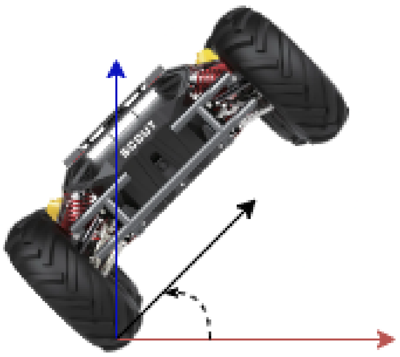
In this section, we delve into the results and discussions stemming from our investigation of a mobile robot with differential kinematics, focusing primarily on its oscillation behavior, navigation architecture, and the innovative plugin designed to address and monitor oscillatory movements. Our empirical observations and data analyses provide valuable insights into the dynamics of robot oscillation and the measures implemented to ensure its stability during missions. Additionally, we explore the nuances of autonomous navigation and the critical role played by recovery and navigation behaviors. This section culminates in a comprehensive examination of the plugin architecture, elucidating how it harnesses IMU data to safeguard the robot's performance, thereby advancing the field of mobile robotics.

3.1 Oscillation Behavior

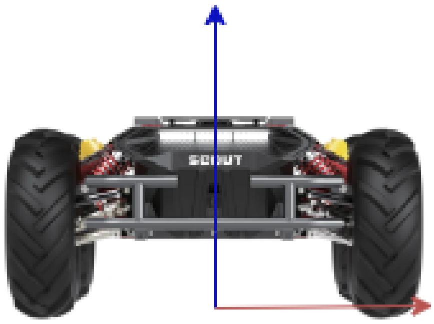
When a mobile robot with differential kinematics rotates around its own axis, it was empirically observed that the robot exhibits an oscillatory movement. Additionally, through empirical analysis, it was noticed that the intensity of oscillation increases with the robot's friction with the ground.

Figure 1 - angulations of the robot

(a) Inclined robot



(b) Robot parallel to ground



(c) Robot in a ramp

Font: author.

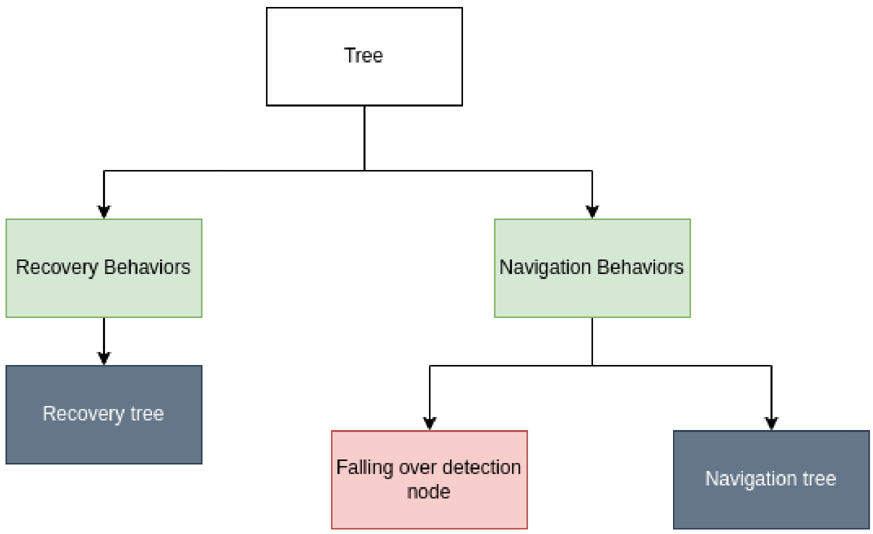
To analyze the IMU data effectively, it is crucial to understand how the oscillation occurs and identify potential exceptions. The robot's angle varies when it tilts, as shown in Figure 1a-b, but this variation also happens when the robot goes up ramps or encounters uneven terrain, as illustrated in Figure 1c. However, for our robot (Scout), the working angle should not exceed 30 degrees.

The key distinction between robot oscillation and its normal movement in the environment lies in the timing of IMU angle measurements. While the robot navigates the environment, there is a gradual change in the robot's angle, which remains stable for an extended period with minimal variation. On the other hand, during shaking movements, there is a drastic angle change within a short period of time. To address this issue, it is essential to analyze not only the angle data from the IMU but also consider the sampling time.

The algorithm leverages the angle data analyzed over time, which represents its derivative, known as angular velocity. The process involves collecting a certain number of samples and calculating their average to reduce the risk of noise leading to false alarms. Afterward, the algorithm compares the obtained value with a pre-established threshold and returns to a stable state if necessary. This way, the robot's oscillation is continuously monitored during mission execution.

3.2 Navigation Architecture

Figure 2 - navigation diagram



Font: author.

The navigation can be executed as exemplified in Figure 2. Autonomous navigation generally involves two main behaviors: "recovery behaviors" that trigger actions when the navigation fails, and "navigation behaviors"that guide the robot towards specific goals. The recovery tree and the navigation tree represent the remaining actions taken by their respective branches. The plugin designed to prevent tipping should be executed while the navigation tree is active, ensuring constant monitoring of IMU data and preventing potential tipping of the robot.

3.3 Plugin architecture

The plugin's architecture is designed to monitor and address the oscillatory movements of the mobile robot (Scout) with differential kinematics. It starts by acquiring data from the IMU (Inertial Measurement Unit) mounted on the robot, which provides information about the robot's orientation and angular velocity. The raw IMU data undergoes signal processing to filter and preprocess it, obtaining clean and accurate angular velocity measurements. The calculated angular velocity is then compared with a pre-established threshold, representing the maximum allowable angular velocity without being considered in an oscillatory state. If the threshold is exceeded, the plugin takes necessary corrective actions to stabilize the robot by generating a signal to the behavior tree. The behavior tree by itself, can be programmed to react by adjusting control parameters like, for example reducing rotational speed. To avoid false alarms, the plugin aggregates data over time, calculating the average angular velocity over multiple measurements. Integrated into the robot's control system and mission execution logic, the plugin continuously monitors the robot's behavior and ensures optimal performance during missions. Logging functionality keeps track of the robot's oscillatory behavior and corrective actions taken, aiding post-mission analysis and debugging.

4. CONCLUSION

The investigation into the oscillation behavior of the mobile robot with differential kinematics revealed important insights into its movement patterns and potential challenges. Through empirical analysis, it was observed that oscillations occur due to the robot's rotation around its axis and are influenced by the friction with the ground. This understanding laid the foundation for designing an effective plugin to address this issue.

The developed plugin, which analyzes IMU data over time, proved to be successful in mitigating oscillation-related problems. By calculating the angular velocity through a series of samples and comparing it against a predefined threshold, the plugin effectively monitored the robot's oscillation during mission execution. Its ability to differentiate between normal navigation and oscillatory movements ensured that false alarms were minimized, thus enhancing the robot's performance and safety during its missions.

In conclusion, the combination of analyzing IMU data over time, employing behavior trees for navigation, and implementing a tipping prevention plugin proved to be a promising solution for addressing oscillation-related challenges in mobile robots. The developed system showcases potential for further enhancements and applications in various robotic missions, contributing to safer and more reliable autonomous navigation.

Acknowledgments

The authors would like to acknowledge SENAI CIMATEC for the support in this research.

5. REFERENCES

[1]ROBOTICS - Wikipedia. <https://en.wikipedia.org/wiki/Robotics>. (Accessed on 07/22/2023).

[2]ROBOTICS Essay | Essay on Robotics for Students and Children in English - A Plus Topper. <https://www.aplustopper.com/robotics-essay/>. (Accessed on 07/22/2023).

[3]ESSAY on Robots in the Future - Free Essay Example - Edubirdie. <https: //edubirdie.com/examples/essay-on-robots-in-the-future/#citation-block>. (Accessed on 07/22/2023).

[4]BISWAS, A.; WANG, H.-C. Autonomous vehicles enabled by the integration of iot, edge intelligence, 5g, and blockchain. Sensors, v. 23, n. 4, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/4/1963>.

[5]VAINIO, M. et al. Safety Challenges of Autonomous Mobile Systems in Dynamic Unstructured Environments: Situational awareness, decision-making, autonomous navigation, human- machine interface. 2020.

[6]THE future of robotics: How will robots change the world? - FutureLearn. <https://www.futurelearn.com/info/blog/introduction-robotics-future-robots>. (Accessed on 07/24/2023).

[7]SOFMAN, B. Online learning techniques for improving robot navigation in unfamiliar domains. [S.l.]: Carnegie Mellon University, 2010.

[8]PERILLE, D. et al. Benchmarking metric ground navigation. In: IEEE. 2020 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). [S.l.], 2020.

[9]GUASTELLA, D. C.; MUSCATO, G. Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review. Sensors, MDPI, v. 21, n. 1, p. 73, 2020.

[10]SáNCHEZ-IBáñEZ, J. R.; PULGAR, C. J. Pérez-del; GARCíA-CEREZO, A. Path planning for autonomous mobile robots: A review. Sensors, v. 21, n. 23, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/23/7898>.

[11]DON'T Let Your Robot Fall Down the Stairs! SICK USA BLOG. <https://sickusablog.com/dont-let-robot-fall-stairs-negative-obstacle-detection-using-lidar-must-mobile-robots/ >. (Accessed on 07/24/2023). Citado na página 2.

[12]OLIVEIRA, L. F.; MOREIRA, A. P.; SILVA, M. F. Advances in forest robotics: A state-of-the-art survey. Robotics, MDPI, v. 10, n. 2, p. 53, 2021.

[13]PAEZ-GRANADOS, D.; BILLARD, A. Crash test-based assessment of injury risks for adults and children when colliding with personal mobility devices and service robots. Scientific reports, Nature Publishing Group UK London, v. 12, n. 1, p. 5285, 2022.

[14]MACENSKI, S. et al. Robot operating system 2: Design, architecture, and uses in the wild. Science Robotics, v. 7, n. 66, p. eabm6074, 2022. Disponível em: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>.

[15]MACENSKI, S. et al. The marathon 2: A navigation system. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.l.: s.n.], 2020.

[16]GITHUB - BehaviorTree/BehaviorTree.CPP: Behavior Trees Library in C++. Batteries included. Https://github.com/BehaviorTree/BehaviorTree.CPP. (Accessed on 07/30/2023).