1. Ala Eddine DJOKHRAB[1,2], 2. Hatem GHODBANE[2], 3. Abdelmalik TALEB-AHMED[3]

University Echahid Hamma Lakhdar – El-oued – BP 789 El-oued, Algeria (1)
Department of Electrical Engineering, Energy Systems Modeling (MSE) Laboratory, Mohamed Khider Biskra University, Algeri (2)
Laboratory of IEMN DOAE. UMR CNRS 8520, University of Valenciennes, 59313 Valenciennes, France (3)
ORCID: 1. 0000-0001-8726-7127

doi:10.15199/48.2024.12.46

Control system of a humanoid robot (NAO) by a laser scanner

***Abstract**. The paper discusses an algorithm that assists in controlling the NAO humanoid robot through a system equipped with a two dimensional laser scanner. By utilizing the external laser scanner in pre-defined confined spaces, the robot's movements become much more reliable. The algorithm is implemented and tested in this system, overseeing the NAO robot's movement, merging data from the robot's built-in sensors with the computational capabilities of the control system and laser scanner data. This system facilitates fast and effortless localization and navigation of one or multiple robots in a feature-defined environment.*

Streszczenie.** Artykuł omawia algorytm wspomagający kontrolę robota humanoidalnego NAO poprzez system wyposażony w dwuwymiarowy skaner laserowy. Wykorzystując zewnętrzny skaner laserowy w predefiniowanych zamkniętych przestrzeniach, ruchy robota stają się znacznie bardziej niezawodne. Algorytm jest implementowany i testowany w tym systemie, nadzorując ruch robota NAO, łącząc dane z wbudowanych czujników robota z możliwościami obliczeniowymi systemu sterowania i danymi z skanera laserowego. Ten system ułatwia szybką i bezproblemową lokalizację oraz nawigację jednego lub wielu robotów w środowisku zdefiniowanym cechami. (**System sterowania robotem humanoidalnym (NAO) za pomocą skanera laserowego)

Keywords: humanoid robots (NAO), laser radar, collision avoidance, robot control.
Słowa kluczowe: roboty humanoidalne (NAO), radar laserowy, unikanie kolizji, kontrola robota.

Introduction

The French company Aldebaran Robotics designed a humanoid robot named NAO (see Gouaillier et al. (2009)) which is capable of complete programming using the custom OS system NAOqi Softbank and the development environment Choregraphe developed [1] [2]. Interaction between humans and the robot can occur in several ways, such as using touch sensors or recognizing sound and images. The robot also contains many useful sensors during movement, such as limit switches on the front part of the feet, two interchangeable cameras with partially overlapping coverage, image processing algorithms that can be processed using pre-defined integrated units, and two ultrasound sensors that allow it to estimate the distance of obstacles in its environment (Fig.1).

To obtain data surrounding the robot, it can be done using motion-incorporated sensors such as ultrasonic sensors, bumpers, vision systems, or data from gyroscopic sensors. In this research paper, we propose using a LIDAR (Light Detection and Ranging) scanner device capable of visualizing the spatial boundaries of the robot's environment. Additionally, we propose an algorithm to process the readings from the LIDAR and merge the acquired knowledge with the robot's sensors that are being implemented, in order to build a joint decision to be passed to the robot's control system [2].

The use of a server addresses the limited computational resources problem at the Android platform level. The built-in sensors provide feedback that may not be accurate enough for independent and safe navigation in a defined and non-empty space, especially if the main goal is to reach a specific location. Therefore, the proposed algorithm relies on taking the simplest possible means to significantly increase the navigational capability of the NAO robot or any other mobile humanoid robot.

Collision avoidance algorithm to identification the Free zone

The general design of the algorithm presented in this paper aims to achieve: identifying the free area and the area constrained by obstacles or barriers such as walls or humans, and tracking the movement of the robot and determining its actual location within the environment, even if the robot is not in the LIDAR's field of view.

Knowing the layout and areas with obstacles and those without obstacles is a very important point for a mobile robot in order for its movement to be fast, its performance more dynamic and smoother. On the contrary, if this robot enters a specified area with obstacles, caution must be taken in its movement to ensure safety, and this is done through modifying the movement algorithm and strategy.

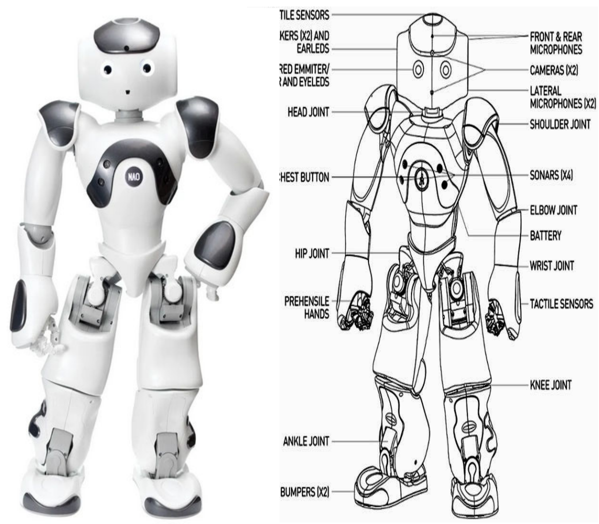


Fig.1. The NAO humanoid robot and some of its sensory devices and descriptive data used during movement.

In the NAO robot, we find many sensors that attract all the information about the robot's environment. However, these sensors have significant limitations.

The ultrasonic sensors can detect obstacles, but do not provide any information about the size or shape of the object. In contrast, the vision system can only recognize simple objects due to its sensitivity to lighting conditions, long response time, limited efficiency, and high time- and CPU-power consumption. These limitations are a result of the hardware and processing power of the built-in vision subsystem. To overcome these limitations, an external computational resource can be used to run advanced scene analysis and feature extraction algorithms, which would lead to improved outcomes.

Despite the presence of bumpers located in the feet of the NAO robot, they should only be used as a last resort to avoid collisions. Unfortunately, they trigger an event too late, once the collision has already occurred, and thus cannot prevent its consequences. Although the system can detect the collision eventually, it does not have sufficient time to react and prevent it from happening.

This is the Pepper Gardiki robot and others (2017)[3], successor to the NAO robot also produced by SoftBank Robotics, and is equipped as standard with laser scanning devices, which enable it to detect and accurately identify its surrounding environment. We did not study it as it is not currently available to us. Instead, we attached an external device with a rotating laser unit for this research, as the NAO robot's devices cannot be modified. This is also an easy option for adjusting and sensing the locations of objects in the experimental area.

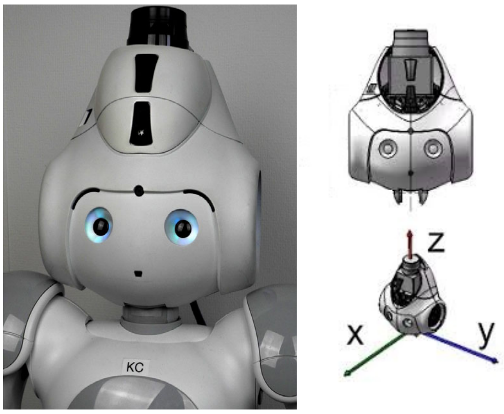
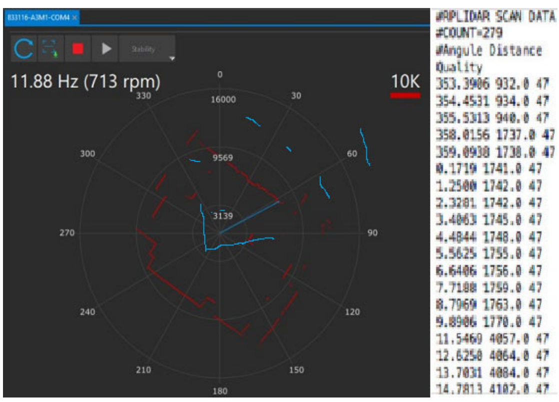


Fig.2. The NAO humanoid Laser head unit Gomez (2017)

Fig.3. Items in the RPLIDAR Development Kit



Free Zone Identification Algorithm

Using data from Lidar, we will propose an algorithm that identifies free zones as follows:

1- Primary stage:
- Save the initial scene state from LIDAR.
- Identify permanent obstacles, whether stationary or moving.
- Robot's initial movement using LIDAR data to determine its position.

2- Runtime loop:
- Reload and save the scene from LIDAR. Verify obstacle identification and define safe movement areas.
- Recheck if the planned movement enters within the safe movement area.
- Inform the system of the change in location and start the movement process, executing the action.

When applying image processing algorithms to preprocessed LIDAR data, scene analysis is performed.

The numbers shown in Figure 13 illustrate the various stages of the algorithms' operation. Figure 4 illustrates the process of identifying permanent obstacles, which is done only once when setting up a new workshop or operating the robot for the first time. The purpose is to verify the knowledge of obstacle boundaries acquired from LIDAR and to define the permanent boundaries. It is useful to define primary safe areas and secondary uncertain movement areas.[4][5]

The next important stage is to link initially separate knowledge domains, which consist of data from the laser scanner and data from distance measurement methods in the robot. The LIDAR-based approach is usually replaced by other methods; however, we will use LIDAR data as an additional source of information because our laser scanner is external (mounted in the room, not on the mobile platform). Although the LIDAR-based approach is accurate and advanced, the necessity for robots to collaborate in the same workshop requires it. This allows robots to obtain the area map by connecting to the external system at any time.

They can also share their location information (or modify it in the system), which synchronizes the LIDAR "map."
After the negotiation stage, when the robot is "aware" of its position within the map, two types of areas can be determined:
- Safe area: Clearly visible to the LIDAR device, the robot is near or within this area, and all obstacles within the area, including walls and furniture, are treated with a safety margin. Moving objects are treated with a larger safety margin.
- Limited movement safety zone: Not visible to LIDAR, but the layout suggests that this area can be used by a robot for navigation.

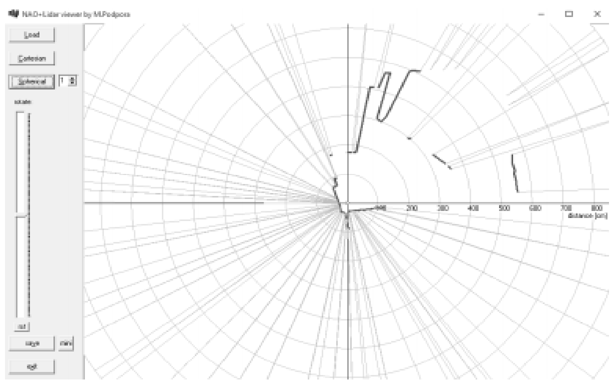


Fig.4. Screenshot of the LIDAR data preview application showing a spherical preview of the raw data.

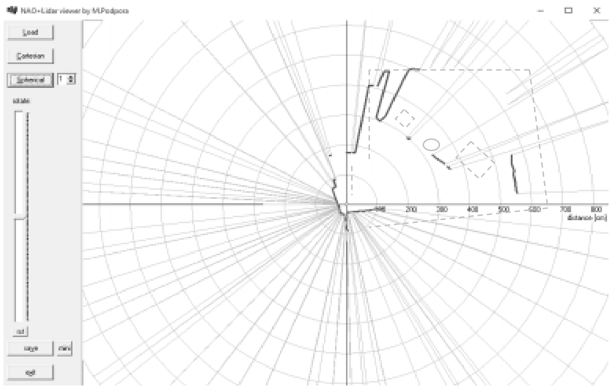


Fig.5. The LIDAR data is superimposed onto predefined room boundaries and obstacles, aiding in robot localization and orientation.

Algorithm for approximating the robot's position

Determining the robot's position from LIDAR data contributes to the quality and accuracy of the robot's location. It also depends on the accuracy of the information, such as the distance between the laser scanner device and other elements of the scene, including the robot. For this reason, determining the robot's position from LIDAR data is crucial in knowing the precise location of the robot within the artificially generated map by LIDAR.

The synchronization algorithm is usually executed only once, during the initialization/setup stage. After connecting to the external system and loading boundaries of LIDAR data, the robot's algorithm compares these boundaries to check if there are any moving objects. Then, if the robot's ultrasonic sensors indicate some free space in front of the robot, the robot makes a short-distance move. During this move, LIDAR data is analyzed for changes, and when the robot stops, it reports its new location to the system, alerting other robots of its presence. The robot stores and updates its location in its memory and in non-volatile memory. Theoretically, it is also possible to determine the locations of charging units.

After the synchronization stage, the actual movement of the robot can result in noticeable changes in LIDAR data, although it is possible to see other moving objects that have been identified within the synchronization algorithm, of course. In the "map preview," the robot's location is marked by a circle with a diameter of 24 pixels (which means a diameter of region A of 0.19 meters and a diameter of region B of 0.5 meters) - this circle includes the circumference of the robot's physical construction and an additional 0.155-meter safety zone, which serves as a correction for errors in robot positioning (Fig. 6). This 0.155-meter margin is particularly useful during the rotation of a humanoid robot.

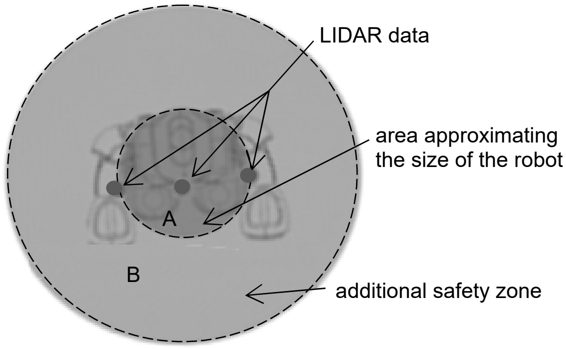


Fig.6. NAO localisation and zones based on its size and position

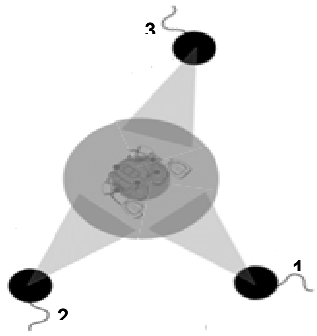


Fig.7. NAO positions during accuracy tests regarding the location of LIDAR.

The proposed robot positioning algorithm is primarily based on LIDAR data, which enables us to at least identify the robot's location. Typically, in the proposed algorithm, these readings form a circle that allows us to quickly and maximally know the robot's position. In the case of multiple robots in the scene, their locations and safety zones will additionally affect the shape of the safe-movement area and the limited-movement area. In Table 1, we summarized the results of determining the robot's position based on the proposed algorithm and compared it to the actual robot location. These errors appear to be acceptable initially, given the ambiguity of the robot's location based on LIDAR data.

Table 1. The Refining the Robot Localization Accuracy Using LIDAR Data Relative to Robot Position (Absolute Error Value)

Distance from LIDAR[m]	Error 1 [mm]	Error 2 [mm]	Error 3 [mm]
0.5	11	4	14
1	17	11	11
1.5	16	5	12
2	16	4	17
2.5	14	13	16
3	19	18	22

Obstacle Avoidance Algorithm

If the robot can navigate from the starting point A to the target point B without leaving the safe-movement area, it is done at normal or increased speed. The robot can also use the map of the area to determine the optimal path, where the optimal path is one that is energy-efficient, time-efficient, or optimal in distance, and any other criterion deemed optimal.

If the robot is forced to navigate outside the safe-movement area, the movement is carried out with increased caution, reduced speed, and ensuring the highest priority and update rate for its ultrasonic sensors.[5][6][7]

If the robot's sensors detect an obstacle that prevents the planned navigation task, the route is replanned considering the new restrictions. All moving objects, whether human or another moving robot, are treated with a double safety zone (Fig. 8).

practical application of this use case

Working with LIDAR data involves processing a series of measurements and converting them into a spherical 2-dimensional coordinate system. The process may be challenging, as depicted in (Fig. 14), but the code and experience gained from implementing the viewer in (Fig. 4 and 5) can be directly applied to the final system, which interfaces the external LIDAR with navigating robots in the area.

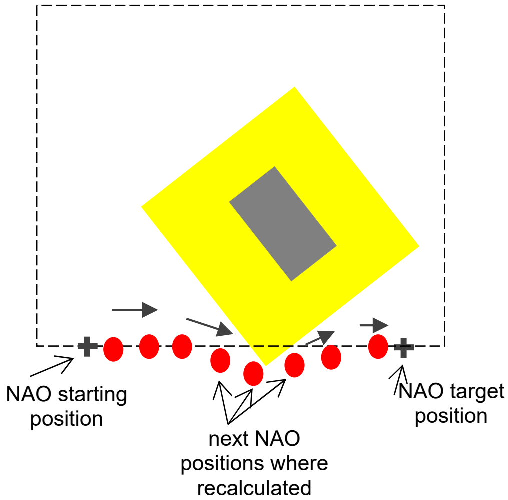


Fig.8. collision avoidance algorithm operations

Once the system is prepared, it is capable of receiving connection requests from robots, transmitting the current data frame, and receiving the current location/position information of the robot. One of the convenient data transmission formats is JSON, although other formats are also acceptable. In a specific example, the NAO robot establishes a connection with the server using a static IP address. It then employs a Python script to retrieve the JSON data of the most recent frame from the server. The robot executes a short-range movement (if feasible) to observe changes in the LIDAR data, aiming to calibrate its position within the LIDAR map. This "location negotiation" algorithm was discussed in a previous section. The robot's location is sent to the server to include it in the JSON structure in case more robots connect to the server and navigate within the same area.[8]

In the preview, the robot's position is represented as a circle with a diameter of 24 pixels. In the server's broker module, the robot's position is depicted as the center of the circle, along with alterations in the shapes of the safe and limited movement areas. The size of the circle in the preview, as well as the subtracted part of the limited movement area in the "areas" data, is approximately equivalent to the outer boundaries of the robot's physical construction in a standing position (Fig. 4 and 6).

The robot obtains information from the server in one of the following forms or structures:
- raw LIDAR data,
- LIDAR data + permanent walls and obstacles,
- LIDAR data + permanent walls and obstacles +locations of the robot(s),
- primary zone (safe-movement-area) data,
- secondary zone (limited-movement-area) data.

In the practical application of this use case, route planning was not integrated into the external system. However, incorporating such functionality would be advantageous, as it would allow for the synchronization of multiple robots' routes to prevent collisions. Implementing route planning directly on the robot introduces the possibility of encountering other robots along its path, necessitating special handling.

Nevertheless, while the robot's route synchronization with LIDAR map data or the system's area data does not offer the ultimate navigation solution due to areas not fully covered by the LIDAR and areas requiring increased precaution, it is essential to consider the robot's odometry system in decision-making and safety algorithms.

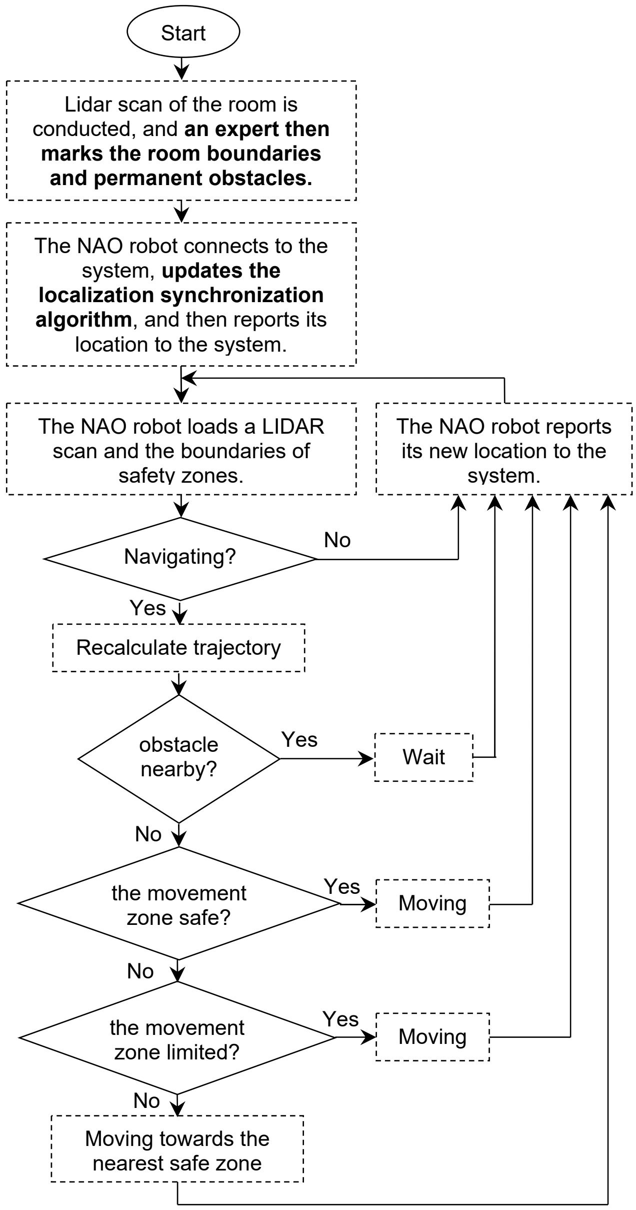


Fig.9. A sketch of the general algorithm.

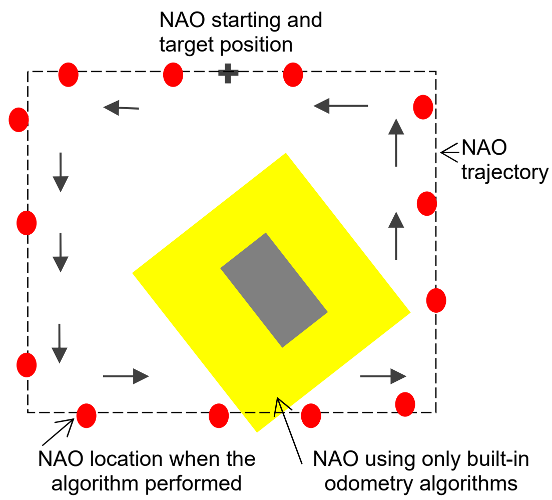


Fig.10. The robot, the obstacles and its movement trajectories.

Movement control system

In this section, we attempt to present the robot control algorithm, which is programmed for specific motion and execution. This presents the robot control algorithm designed in a specific execution form.

This is to illustrate the possibility of modifying data in all units, and then to explain the mechanism of determining the algorithm of the workshop map that the robot is supposed to move in, such as room dimensions, LIDAR range, LIDAR mounting height, robot cross-section size, and so on.[7][9]

Also, among the benefits we gain from programming in a specific execution form is the ability to choose different and alternative methods to accomplish some tasks, such as collision detection methods, strategies for determining new movement trajectories, and others.

In order to evaluate the efficiency of the proposed algorithm, the below mentioned tests were conducted. Their goal was to show if the use of the proposed system woul significantly improve the precision in moving robot to a location or not (Fig. 12).

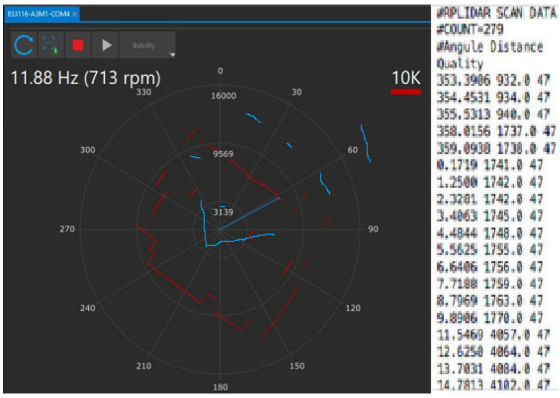


Fig.11. Initial data acquired from LIDAR

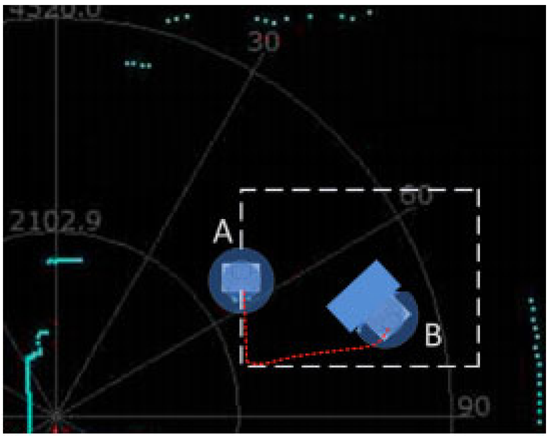


Fig.12. The robot, the obstacles and its movement trajectories visualized on top of the LIDAR preview

The path tracking accuracy of NAO robot is poor (e.g. Wen et al. (2014) and Wei et al. (2014)). The errors of reaching a given location even at a distance of 1 m can be significant. The reason for them is the composition of several factors:

varying friction and unevenness of the floor surface, asymmetric movements of the left and right legs of the robot during walking, angular precision error when rotating the robot, looseness of the robot's construction, etc. The severity of these issues may be analyzed and evaluated individually, but in general they can can also be treated as random. An example of problems with completing a test route using only the built-in odometry mechanisms is presented in (Fig. 12).

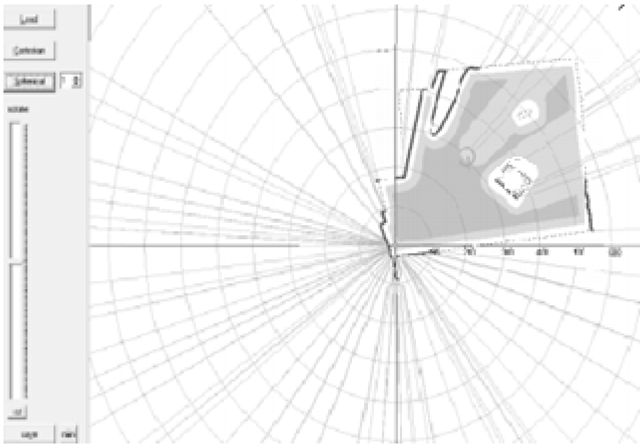


Fig.13. Safe and Limited the movement regions are defined.

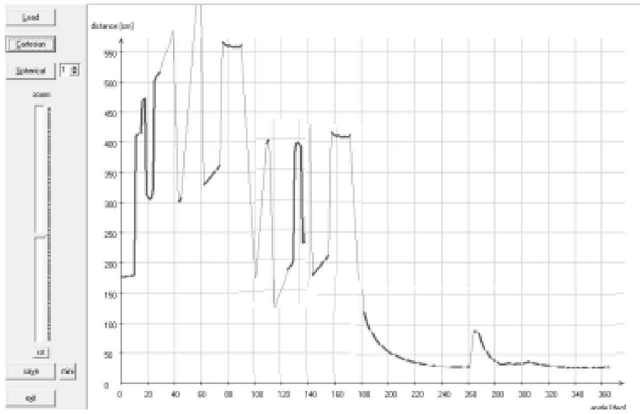


Fig.14. Preview of Raw LIDAR Data in Cartesian Coordinates.

Due to the utilization of solely the built-in odometry mechanisms for navigation, none of the 4-8 attempts illustrated in (Fig. 12) were successful. Throughout the turnover maneuver, the NAO robot introduces a notable error; failure to compensate for this error resulted in the robot becoming stuck upon colliding with the obstacle.[8][9]

When we used the algorithm proposed by us, despite the robot passing the test track every time (Fig. 10), it did not exceed a positioning error of 3.9 cm after reaching the starting position, as shown in Table 2 for the results of five attempts to navigate from starting point A and return to the same point via the specified path.

Table 2. Difference Between the Starting and Destination Point of the Robot (Trajectory – Fig. 10)

Attempt	1	2	3	4	5
|Accuracy|(cm)	3.9	3.1	3.9	3.4	2.8

Futre work

The proposed algorithm and system concept can be used in conjunction with other sources to receive information about the robot's location and obstacles, such as an external video system similar to distributed computer vision systems, to provide data and maps in a more convenient and accurate manner to facilitate the detection of moving obstacles, especially in areas with limited freedom of movement.

Additionally, a type of filter, such as a Kalman filter The

Study of Fractional order Controller with SLAM in the NAO Robots, can be incorporated into the algorithm to provide a satisfactory addition to the algorithm.

Furthermore, it is possible to expand the system's capabilities by using more laser sensors to obtain more information. In this case, the area of limited movement will be much smaller.

Conclusions

The proposed algorithm significantly improves the reliability in determining the safe movement area for the robot, despite having some limitations and drawbacks. Additionally, it enhances efficiency in tasks involving reaching a specific location compared to built-in standard algorithms. Furthermore, it can be extended by adding more laser sensors or integrating with a video vision system for application in other mobile robots.

Transferring some calculations to an external system instead of performing them on the robot may allow for the use of online computational processes, which in turn are cost-effective. This proposal is now widely accepted, especially in the field of mobile robots.

Authors: *Ala Eddine Djokhrab, University Echahid Hamma Lakhdar BP 789 El-oued, Algeria; Email: ali-djokhrab@univ-eloued.dz; Prof. Hatem Ghodbane,University of Mohamed Khider Biskra, Algeria, Department of Electrical Engineering; Email: Ghodbane_hatem@yahoo.fr; Prof. Abdelmalik Taleb-Ahmed, University of Valenciennes, 59313 Valenciennes, France; Email: taleb@univ-valenciennes.fr*

REFERENCES

[1] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, B. Maisonnier. *Mechatronic design of NAO humanoid*. In: 2009 IEEE International Conference on Robotics and Automation, 2009. SoftBank Robotics, NAOqi Developer guide.

[2] E. Pot, J. Monceaux, R. Gelin, B. Maisonnier, Choregraphe. Choregraphe: *A Graphical Tool for Humanoid Robot Programming*. Robot and Human Interactive Communication, IEEE, 2009. SoftBank Robotics, Laser head v.3.2.

[3] N. Gomez. *Inside NAO*, 2017.

[4] A. Gardecki and M. Podpora. *Experience from the operation of the Pepper humanoid robots*. Progress in Applied Electrical Engineering (PAEE), 2017.

[5] Slamtec Co. *Lt., RPLIDAR A2 Laser Range Scanner*.

[6] S. Wen, K.M. Othman, A.B. Rad, Y. Zhang, Y. Zhao. *Indoor SLAM Using Laser and Camera with ClosedLoop Controller for NAO Humanoid Robot. Abstract and Applied Analysis*, Hindawi, 2014.

[7] C. Wei, J. Xu, C. Wang, P. Wiggers and K. Hindriks. *An Approach to Navigation for the Humanoid Robot Nao in Domestic Environments. TAROS 2013: Towards Autonomous Robotic Systems*, pages 298–310, Springer, 2014.

[8] A. Gardecki and M. *Podpora. Extending vision understanding capabilities of NAO robot by connecting it to remote computational resource*. Progress in Applied Electrical Engineering (PAEE), 2016.

[9] R.J. Radke. *A Survey of Distributed Computer Vision Algorithms*. Handbook of Ambient Intelligence and Smart Environments, Springer, 2010.

[10] J.-G.Lu,Y.Chen, andW.Chen, *Robust asymptotical stability of fractional-order linear systems with structured perturbations*, Computers and Mathematics with Applications, vol. 66, no. 5, pp.
873–882, 2013.

[11] NAO Technical Overview. Available online: http://doc.aldebaran.com/2-1/family/robots/masses_robot.html (accessed on 20 September 2022).

[12] Available online: https://www.aldebaran.com/en/pepper (accessed on 20 September 2022).

[13] Pino, O.; Palestra, G.; Trevino, R.; De Carolis, B. *The Humanoid Robot NAO as Trainer in a Memory Program for Elderly People with Mild Cognitive Impairment*. Int. J. Soc. Robot. 2020, 12, 21–33.

[14] Jim énez, M.; Ochoa, A.; Escobedo, D.; Estrada, R.; Martinez, E.; Maciel, R.; Larios, V. *Recognition of Colors through Use of a Humanoid Nao Robot in Therapies for Children with down Syndrome in a Smart City*. Res. Comput. Sci. 2019, 148, 239–252.

[15] MILOUDI, K., MOULAI, H., and AZIZI, H. (2023). *Modeling and Numerical Simulation of Eddy Current Sensors for Electromagnetic Characterization of Fluids*. Przeglad Elektrotechniczny, 2023(10).

[16] Robaczewski, A.; Bouchard, J.; Bouchard, K.; Gaboury, S. Socially Assistive Robots: *The Specific Case of the NAO*. Int. J. Soc. Robot. 2021, 13, 795–831.

[17] Franke, A.; Nass, E.; Piereth, A.-K.; Zettl, A.; Heidl, C. *Implementation of Assistive Technologies and Robotics in Long-Term Care Facilities: A Three-Stage Assessment Based on Acceptance, Ethics, and Emotions*. Front. Psychol. 2021, 12, 694297.