A Novel Docking Algorithm Based on the LiDAR and the V-shape Features

Sercan Çağdaş Tekkök [1], Bekir Bostancı [2], Mehmet Emre Söyünmez [3], Pınar Oğuz Ekim[4*]

[1*] İzmir Ekonomi Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, İzmir, Türkiye, cagdastekkok@gmail.com
[2] İzmir Ekonomi Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, İzmir, Türkiye, bekirbostanci@gmail.com
[3] İzmir Ekonomi Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, İzmir, Türkiye, emresoyunmez@gmail.com
[4] İzmir Ekonomi Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, İzmir, Türkiye, (ORCID: 0000-0003-1860-4526),
pinar.ekim@ieu.edu.tr

Abstract

This paper presents a feature based docking strategy with a LiDAR for a non holonomic mobile platform to be charged at the charging station. The docking station has only a V-shape structure to be detected from the LiDAR scans. The proposed docking algorithm consists of three main sections which are the localization of the V-Shaped feature with the LiDAR measurements, the path generation with a quintic polynomial and the Proportional Integral Derivative (PID) implementation to track the generated points of the path and correct the yaw angle along the way. Once the relative position of the docking station is detected by matching the synthetically generated V-shape feature with the current Lidar scans, the trajectories are generated by calculating the coeeficients of the quintic polynomial. Afterwards, the initial states, the states and the yaw angles at the points along the path are taken as inputs and the algorithm corrects the robot's yaw angle as it follows the path by a PID control. The algorithm is both tested within the simulation environment and the real test environment. The platform docks with a good accuracy which is around $\pm$ 2 cm in *x* and *y*, $\pm$ 1° in θ. With the proposed strategy, additional sensors are not necessary on the platform or on the charging station. Moreover, the robot can navigate even in the absence of light with a good accuracy. Hence, a cheaper solution is obtained for an important problem of autonomous mobile robotic applications.

Keywords: Non holonomic platform, Feature based docking, LiDAR.

V-şekil Özelliklerine ve Lidar Sensörüne Dayalı Yeni Bir Kenetlenme Algoritması

Öz

Bu çalışma, şarj istasyonunda şarj edilecek holonomik olmayan bir mobil platform için LiDAR ile özellik tabanlı bir kenetlenme stratejisi sunar. Şarj/kenetlenme istasyonu, LiDAR taramalarından tespit edilebilecek yalnızca V şeklinde bir yapıya sahiptir. Önerilen yerleştirme algoritması, V-şekil özelliğinin LiDAR ölçümleriyle bulunması, beşli polinom ile yol oluşturulması ve yolun oluşturulan noktalarını izlemek ve düzeltmek için Orantılı İntegral Türev (PID) uygulaması olmak üzere üç ana bölümden oluşur. Sentetik olarak oluşturulan V-şekli özelliği mevcut Lidar taramaları ile eşleştirilerek kenetlenme istasyonunun göreceli konumu tespit edildiğinde, beşli polinomun katsayıları hesaplanarak yörüngeler oluşturulur. Daha sonra, yol boyunca noktalardaki başlangıç durumları, durumlar ve sapma açıları girdi olarak alınır ve algoritma, PID kontrolü ile yolu takip ederken robotun sapma açısını düzeltir. Algoritma hem simülasyon ortamında hem de gerçek test ortamında test edildi. Platform, *x* ve *y*'de $\pm$ 2 cm, θ'de $\pm$ 1° civarında olan iyi bir doğrulukla yanaşmaktadır. Önerilen strateji ile platformda veya şarj istasyonunda ek sensörler gerekli değildir. Üstelik robot, ışık olmadığında bile iyi bir doğrulukla hareket edebilir. Böylelikle otonom mobil robotik uygulamalarının önemli bir problemi için daha ucuz bir çözüm elde edilmektedir.

Anahtar Kelimeler: Holonomik olmayan platform, Özellik tabanlı kenetlenme, LiDAR.

* Corresponding Author: pinar.ekim@ieu.edu.tr

1. Introduction

Mobile robots have the ability to travel on all free spaces of an environment. The flexibility is achieved through more complex navigation systems and more powerful sensors. A particular problem raised upon the use of mobile robots is the docking procedure, where the three variables (position and orientation) have to be controlled in a 2D space. The automatic recharging, just one function of the docking station, is essential for achieving a long-term autonomy (Zhang et al., 2018). There are several docking methods which use different sensors and algorithms in the literature as explained in the following paragraphs.

Some of the previous works exploited infrared (IR) emitter-receiver pairs to align their modules for docking due to its low-cost, small size and low power consumption. The docking mechanism deploys th Extended Kalman Filter (EKF) to fuse the measurements from IR and encoders to estimates the orientation and distance. After that these informations are used to align two modules in order to connect (Won et al., 2015). The authors of (Mira Vaz et al., 1997) installed two infrared scanners on the mobile platform to be able to detect the passive reflectors mounted on the docking station. The location of the mobile robot is always taken relative to the docking station by exploiting the triangulation. A similar method with two IR detectors attached to the docking station was proposed in (Song et al., 2015). A surveillance robot moves parallel to the docking station until two IR sensors detect the robot and then the robot rotates 90 degree to face with the station. Five IR LEDs, which are installed in the docking station, divide the docking area into nine regions (Kim et al., 2005). Additionally, the docking station is composed of a toggle switch and of two clasps to improve the robot posture which approaches with an offset and an angle error. Six IR sensors used by the receiver are located at the front and the side of the robot. Kobuki's docking algorithm uses three IR receivers on the TurtleBot and three IR-Emitters on the docking station (Testing automatic docking, 2021). The IR emitters on the docking station divide the docking field into three regions. And for each region the near field and the far field are described. The infrared sensors are cheap but they are prone to errors due to the IR beam reflection and the ordinary day light.

Vision based docking systems are another solution. The object recognition algorithms were integrated with the navigation control strategies in (Kartoun et al., 2009). A visual tag, which is placed on the docking station, is detected by the vision based docking system (Santos et al., 2017). The tag relative position and orientation is obtained and then the robot's trajectory is calculated for the docking approach. A cubic Bezier Curve was used in which the starting point is the position of the robot at the moment when it saw the tag and the end point is the position of the tag at that every moment. In the same article, the authors compared the performance of the docking algorithm with different tags. A drawback of the traditional vision based solutions is that the necessity of light.

A LiDAR sensor can be used with or without day light. The authors of (Rosa et al., 2014) exploits Censi's Canonical Scan Matcher (Censi 2008) with a pretty expensive LiDAR. Another docking algorithm, which uses features as a landmark, was proposed in (Rowe, 2020). To extract the dock, the points that belong to the dock is identified by using Euclidean clustering using nearest neighbours. Once the raw point clouds have been divided into clusters, each cluster is processed to extract any existing lines by using RANSAC. The parametrized dock model and the detected lines are matched by searching through all clusters. After that the iterative closest point algorithm is applied to find the transformation between the robot's coordinate frame and the dock's coordinate frame.

Fusing the information from different sensor types is employed in the literature. For example, the direction information is obtained from an IR sensor and the distance information from an ultrasound sensor (Sreenivas M. V. And Shivakumar M., 2019). The authors of (Quilez et al., 2015) proposed using QR code landmarks and IR distance sensors to enable the autonomous docking. The datamatrix is physically attached to a wall, its pose is computed using the calibrated RGB camera (Marostica et al., 2014). The laser scanner is exploited to compute the wall parameters. Later, the motion planner utilizes the matched datamatrix and the scanned wall's line information.

In order to the pins to be in contact with the metal on the base, the positioning of the robot must be very precise and the standard localization and path planning do not offer enough precision for the task. This paper presents a docking procedure for a nonholonomic mobile robot which is based on LiDAR scan data and a V-shape feature at the charging station. The novel methodology combines the trajectory generation and following, together with a LiDAR based localization system. Hence with this strategy, we do not need additional sensors on the platform and, the robot can navigate in the absence of light with a good accuracy.

Section 2 describes the configuration of the experimental set-up. The sensor data and the associated signal processing localization algorithm are described in Section 3. The trajectory generation is represented in Section 4. Extensive experimental tests were conducted, some of which are presented in Section 5. The paper concludes with the presentation of conclusions and directions for the future works in Section 6.

2. The System Configuration

The system consists of two main sensors, a main computer and a micro-controller to process and pass the information to the computer which utilizes Robot Operating System (ROS) for communication. The RPLidar and the odometry data are used as sensors' measurements and the data are collected with ESP32 which passes the information to the main computer with ROS. A V-shaped feature is placed on top of the charging station (Fig. 1), which is located near the wall. Its location is approximately known, from the map data that priorly obtained with G-mapping (Grisetti et al., 2005) to the robot so it can navigate somewhere near to the station to start the docking process. Once the docking is finished both positive and negative terminals of the charging station are expected to contact with the battery terminals of the robot in Fig. 2

Figure 1 Station Picture
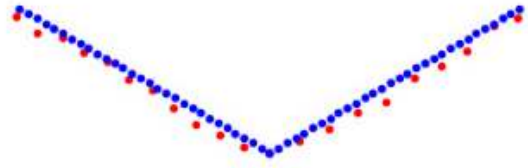

Figure 2 Robot Docking Side

3. Algorithms

The docking algorithm has three main sections which are the localization of the V-Shaped feature with the LiDAR measurements, the path generation with a quintic polynomial and the Proportional–Integral–Derivative (PID) implementation to track the generated points of the path and correct the angle along the way.

3.1. Localization with LiDAR data

To achieve a successful docking, firstly the exact location of the station should be determined, so that a path that leads to it can be generated. This is achieved by LiDAR measurements, which outputs 2D point clouds with 360 samples each refers to $1°$. However, the exact shape of the feature should be known to the algorithm so that it can match that with the measurements of LiDAR and detect the feature. Therefore, a sample feature measurement is generated synthetically with 60 samples on the computer that exactly matches the dimensions of the real life feature which is placed on top of the station. The matching algorithm takes the advantage of euclidean distances to calculate the error between the samples and the measurements. The sample shape is also rotated for $360°$ at each measured point and the matched points are calculated for every iteration. An example of the matched feature and the measurement data can be seen in Fig. 3.
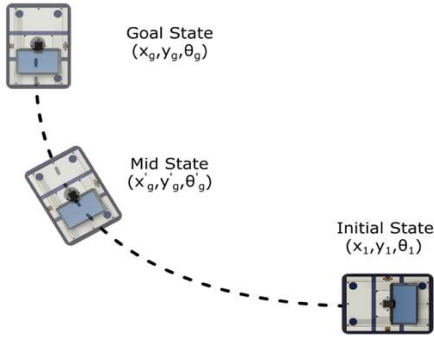

Figure 3 Blue: Reference Feature, Red: LiDAR Scan Data

In order to achieve a more accurate position and angle at the end of the docking, the robot can set the goal to the middle of the trajectory and re-run the localization segment with a higher precision to correct itself. However, since the localization process rotates the reference feature point $360°$ at each one of the 360 LiDAR points, it takes around 35 seconds, which is not very time-efficient. Although it is possible to run the algorithm as it is, some changes can be made to accelerate the localization process at the second step to approximately be 3 seconds (by neglecting the unnecessary points and rotations). To do so it is needed to know where will the feature be, once the robot moves to the midpoint. Therefore, $\vec{r}_{o_2 o_3}$ vector is defined that it gives the coordinates of the feature at the second frame (the mid state in Fig. 4) (Özgören, 2020) as follows

$$\{\vec{r}_{o_2 o_3} = \vec{r}_{o_2 o_1} + \vec{r}_{o_1 o_3} = -\vec{r}_{o_1 o_2} + \vec{r}_{o_1 o_3}\}^{(2)} \tag{1}$$

The heading angle difference between two points is defined by

$$\Delta\theta = \theta_2 - \theta_1 \tag{2}$$

and the transformation matrix $\hat{C}$ is defined by

$$\hat{C}^{(1,2)} = \begin{bmatrix} cos\Delta\theta & -sin\Delta\theta \\ sin\Delta\theta & cos\Delta\theta \end{bmatrix} \tag{3}$$

Once the transformation is applied to the $\vec{r}$ to get them in the initial frame, the following equations are obtained

$$\vec{r}_{o_2 o_3}^{(2)} = -\hat{C}^{(2,1)}\vec{r}_{o_1 o_2}^{(1)} + \hat{C}^{(2,1)}\vec{r}_{o_1 o_3}^{(1)} \tag{4}$$

$$\vec{r}_{o_2 o_3}^{(2)} = \hat{C}^{(1,2)T}(\vec{r}_{o_1 o_3}^{(1)} - \vec{r}_{o_1 o_2}^{(1)}) \tag{5}$$

$$\vec{r}_{o_2 o_3}^{(2)} = \hat{C}^{(2,1)}\begin{bmatrix} x_g - x'_g \\ y_g - y'_g \end{bmatrix} \tag{6}$$

After some simplifications, the coordinates of the goal point is found in (6) where $\hat{C}^{(1,2)T} = \hat{C}^{(2,1)}$.

These neglected features are decided by getting the position and the yaw of the robot from the proposed derivations so the location of the feature will be estimated at the midpoint. Thus, when the robot is at the mid state as in Fig. 4, all the unnecessary LiDAR measurements can be disregarded and the reference feature rotations will be far less than 360 degrees this time instead. Hence the number of iterations will be lowered.

Figure 4 The position and the orientation of the robot at the initial state, the mid state and the goal state

3.2. A trajectory sample for reaching a goal with 2 steps

Once the location of the feature is found with respect to the robot, it is now possible to generate a path to this goal. This will be achieved by the quintic polynomial approach for one dimensional robot motion as described in (Takashi et al., 1989) and it is given as follows

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \qquad (7)$$

where a_0, a_1, a_2, a_3, a_4, and a_5 are the quintic polynomial coefficients. Since the starting and the stopping points are known, the starting position, the velocity, and the acceleration are taken as x_s, v_s and a_s respectively. x_e, v_e, and a_e are the ending position, the velocity, and the acceleration, respectively. At time $t = 0$

$$x(0) = a_0 = x_s. \qquad (8)$$

Differentiating the equation (7), the following formulations are obtained

$$x'(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4, \qquad (9)$$

$$x'(0) = a_1 = v_s. \qquad (10)$$

Then, differentiating the equation (9) with t again,

$$x''(t) = 2a_2 + 6a_3 t + 12a_4 t^2, \qquad (11)$$

$$x''(0) = 2a_2 = a_s. \qquad (12)$$

a_0, a_1 and a_2 can be calculated by using (8), (10) and (12) and the boundary conditions. a_3, a_4, and a_5 are still unknown in (7). Assume that the end time for a maneuver is T, the following equations can obtained from (7), (9) and (11)

$$x(T) = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 + a_5 T^5 = x_e \qquad (13)$$

$$x'(T) = a_1 + 2a_2 T + 3a_3 T^2 + 4a_4 T^3 + 5a_5 T^4 = v_e \qquad (14)$$

$$x''(T) = 2a_2 + 6a_3 T + 12a_4 T^2 + 20a_5 T^3 = a_e \qquad (15)$$

By exploiting (13), (14) and (15), a_3, a_4, and a_5 can be calculated by solving the linear system in (16)

$$\begin{bmatrix} T^3 & T^4 & T^5 \\ 3T^2 & 4T^3 & 5T^4 \\ 6T & 12T^2 & 20T^3 \end{bmatrix} \begin{bmatrix} a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} x_e - x_s - v_s T - 0.5a_s T^2 \\ v_e - v_s - a_s T \\ a_e - a_s \end{bmatrix} \qquad (16)$$

Once all the coefficients are found, it is possible to generate the path as shown in Fig. 5. As a final step, this function will be sampled and each point will be given to the PID as goals so that each iteration angle can be corrected to follow the path which will be discussed in the next section.
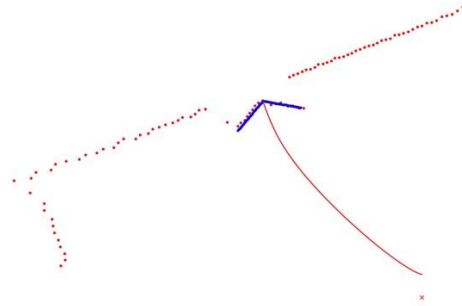

Figure 5 The calculated route, where the blue dots represent the reference pattern, while the red dots and 'x' indicates the LiDAR data and the LiDAR location on the robot respectively.

3.3. PID Path Following

PID controller, which is variously used in control systems, works as a control loop that gives a feedback to adjust and correct the system. After getting the path that needs to be tracked from the previous part, the PID can be initiated. The initial states, and the yaw at the points along the path are taken as inputs and the algorithm corrects the robots yaw angle as it follows the path. Here, the error is calculated at each step as follows:

$$\varepsilon = arctan(\delta_y, \delta_x) - \theta, \qquad (17)$$

where, ε is error between the goal angle and the robot angle, δ_y and δ_x are the differences in x and y positions between the goal and the current states and θ is the current angle. The robot is driven with the linear velocity of 0.08 m/s and the angular velocity is calculated as:

$$\omega = \kappa_P \varepsilon + \kappa_I \varepsilon + \kappa_D e. \qquad (18)$$

Here κ_P, κ_I and κ_D are the PID constants, which are obtained from PID-tuning, ε is the cumulated error and e is the previous error.

4. The Trajectory Generation and Following

The proposed solution to the docking problem consists of locating the feature in the LiDAR measurements, generating the direct path to the designated location with the quintic polynomial approach and following the generated path with a PID control. There are two sensory inputs: the LiDAR information as one shot measurements and the odometry data as continuous input to the system. An example of generated routes can be seen in Fig. 6.
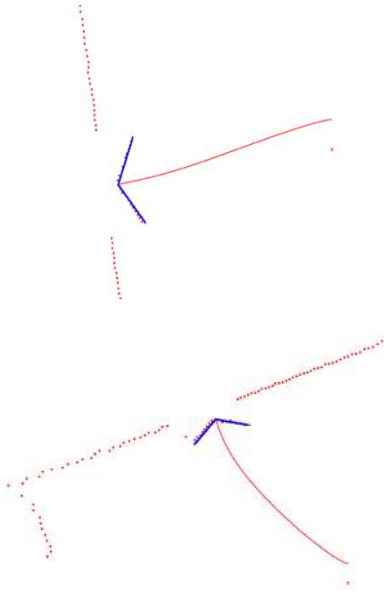

Figure 6 Red curve: Generated path of the algorithm, Blue V-shape: the sample feature matched with measurements, Red dots: LiDAR measurements

The matched sample feature is depicted with blue dots and the LiDAR measurements are given as red dots. The red "x" mark shows the LiDAR's location and the red line shows the generated path which starts at the robot's center next to the "x" mark. After the trajectory is sampled and fed to the PID controller, the robot can finish docking.

5. Results and Discussion

The experiments were conducted with the aforementioned algorithms. With only one step structure without the mid state, the system was able to generate a path to the station and docked with 8 centimeters error and ±3 degrees difference. Although it was fairly accurate, the springs behind the connectors are short and strong and this may cause a connection problem. Therefore, the docking process was splitted into two main steps as explained in 3.1 and 3.2. Each step includes the detection of the V-shaped feature, the quintic path generation and the path tracking. Instead of going straight to the station, the goal now is moved 1 meter in front of the station as shown in Fig. 7 and the first step is repeated at that location as shown in Fig. 8. The second step is resulted in a more accurate docking with approximately 2 centimeters error and ±1 degree difference.
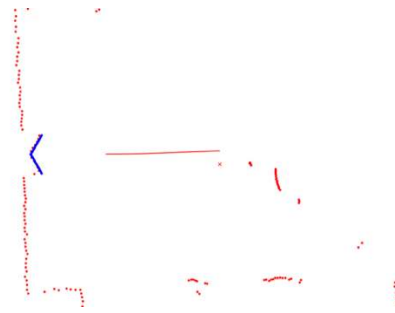

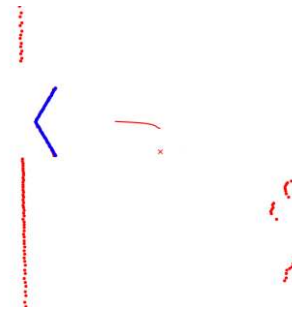**Figure 7** The goal point is set to 1 meter in front of the reference point

Figure 8 The model is executed again with a higher precision and accuracy, the goal point is set 0.5 meter in front of the reference point since in the docked state, the distance between the center of the robot and reference point is 50 centimeters.

6. Conclusions and Recommendations

This paper presented a novel docking method without any extra sensors such as Camera to detect the QR codes or IR. It just exploited on-board sensors which are already used for mapping and navigation of the robot. The experiments showed that the docking with a single step as described in Section 5 results in an inaccurate docking compared to two-step case which includes the same cycle that is done in the first step but it started in almost midway to the goal. Although it adds some delay to the whole process, the delay can be minimised by narrowing down the search area to the previously found feature and search it again around those angles of LiDAR measurements. This speeds up the search process which is the most costly part of the system in terms of the calculation time. As a future work, the model will be optimized in order to acquire better accuracy and speed. Moreover, the precision rate of Section 3.1 will be adjusted according to the distance between the robot and station autonomously.

7. Acknowledge

References

Zhang Z., Meng Y., Song B., Meng X. and Li J. (2018). Design and implementation of an automatic charging system for intelligent patrol robot. *System Science and Control*, vol. 6, no. 3, pp. 19-27, 2018.

Won P., Biglarbegian M., and Melek W. (2015). Development of an Effective Docking System for Modular Mobile Self-Configurable Robots Using Extended Kalman Filter and Particle Filter. *Robotics*, vol. 4, pp. 25-49.

Mira Vaz P., Ferreira R., Grossmann V. and Ribeiro M. I. (1997). Docking of a mobile platform based on infrared sensors. *IEEE Int. Symposium on Industrial Electronics*, July 1997, Guimaraes, Portugal.

Song G., Wang H., Zhang J., and Meng T. (2011). Automatic Docking System for Recharging Home Surveillance Robots. *IEEE Trans. on Consumer Electronics*, vol. 57, no. 2, May.

Kim K. H., Choi H.D., Yoon S., Lee K. W., Ryu H.S., Woo C. K., and Kwak Y.K., 2005. Development of Docking System for Mobile Robots Using Cheap Infrared Sensors. *1st Int. Conf. on Sensing Techonology*, November, Palmerston North, New Zealand.

Testing Automatic Docking. Accessed:2021-02-22, http://wiki.ros.org/kobuki/Tutorials/AutomaticDocking.

Kartoun U., Stern H., Edan Y., Feied C., Handler J., Smith M., and Gillam M., (2006). Vision Based Autonomous Robot Self-Docking and Recharging. *World Automation Congress*, July, Budapest, Hungary.

Santos L., Neves dos Santos F., Mendes J., Ferraz N., Lima J., Morais R., and Costa P., (2017). Path Planning for Automatic Recharging System for Steep-Slope Vineyard Robots. *ROBOT 2017: Third Iberian Robotics Conference, Advances in Intelligent Systems and computing*, vol. 693, Springer.

Rosa S., Russo L. O. and Bona B., 2014. Towards a ROS based autonomous cloud robotics platform for data center monitoring. *Proceedings of the 2014th IEEE Emerging Technology and Factory Automation*, pp. 1-8, Barcelona, Spain.

Censi A., 2008. An icp variant using a point-to-line metric. *IEEE International Conference on Robotics and Automation*, pages 19–25.

Rowe C., 2020. Docking Solution for Autonomous Mobile Robots. *M.Sc. Thesis*, Department of Engineering, Trinity College, Ireland.

Sreenivas M. V. And Shivakumar M., 2019. Sensor Guided Docking of Autonomous Mobile Robot for Battery Recharging. *Int. Jour. of Recent Technology and Engineering*, vol. 8, no. 4.

Quilez R., Zeeman A., Mitton N., and Vandaele J., 2015. Docking autonomous robots in passive docks with infrared sensors and QR codes. *Int. Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities*, June, Vancouver, Canada.

Marostica M., Bullo M., Moro M., Dughiero F., and Menegatti E., 2014. A wireless recharging system for electrical agriculture robots with autonomous docking. *Computer Science*.

Grisetti G., Stachniss C., and Burgard W., 2005. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. *In Proceedings of the 2005 IEEE International conference on robotics and automation (ICRA)*, 2432-2437.

Özgören M.K., 2020. Kinematics of General Spatial Mechanical Systems. Wiley, 2020.

Takahashi A., Hongo T., Ninomiya Y., and Sugimoto G., 1989. Local Path Planning And Motion Control For Agv In Positioning. *IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89), The Autonomous Mobile Robots and Its Applications*, Tsukuba, Japan, 1989, pp. 392-397.