
IMPLEMENTATION OF SLAM NAVIGATION AND PATH PLANNING USING ROBOTIC PLATFORM EQUIPPED WITH LASER SCANNER AND ODOMETRY SYSTEM²²

Eng. Georgi Georgiev, PhD Student

Department of Telecommunications,
“Angel Kanchev” University of Ruse

Tel.: 082 888 353

E-mail: gdgeorgiev@uni-ruse.bg

Assoc. Prof. Georgi Hristov, PhD

Department of Telecommunications,
“Angel Kanchev” University of Ruse

Tel.: 082 888 663

E-mail: ghristov@uni-ruse.bg

Assoc. Prof. Plamen Zahariev, PhD

Department of Telecommunications,
“Angel Kanchev” University of Ruse

Tel.: 082 888 663

E-mail: pzahariev@uni-ruse.bg

Assist. Prof. Diyana Kinaneva, PhD

Department of Telecommunications,
“Angel Kanchev” University of Ruse

Tel.: 082 888 353

E-mail: dkyuchukova@uni-ruse.bg

Abstract: One of the most interesting tasks modern mobile robots are facing with are SLAM navigation and path planning. In order to move freely in space, mobile robots must be familiar with their surroundings. SLAM navigation is a computational problem when creating or updating a map in an unknown environment, while monitoring the position of the robot in it. Initially, this task was unsolvable, as the chicken-egg problem. We currently know several algorithms for solving the problem, at least approximately, in trackable time and for certain environment. SLAM algorithms are based on concepts in computational geometry and computer vision and are used in robot navigation, robot-based cartography and odometry for virtual reality.

Path planning tasks include avoiding walls, moving robot without falling down stairs, and more. The path planning algorithm would take these tasks as input and create speed and rotation commands to send to the wheels of the robot. Motion planning algorithms can manage multi-joint robots and perform complex tasks, such as moving objects.

Two scripts will be created in this report: for SLAM navigation and for path planning. ROS (robot operating system) will be used. The scripts will be executed on a mobile robot equipped with a laser scanner and an odometry system. The success rate of the robot to handle the tasks will be explored.

Keywords: SLAM navigation, Path planning, Laser scanner, ROS, Mobile robot

ВЪВЕДЕНИЕ

Роботите са способни да изпълняват специфични задачи в различни работни среди и намират приложение в различни области като производството, медицината, военните, в космоса, както и в моретата и океаните. Като най-широко използваната категория в областта на роботиката, интелигентните мобилни роботи имат способността да се движат гъвкаво в

²² Докладът е представен на заседание на секция 3.2 на 29 октомври 2021 с оригинално заглавие IMPLEMENTATION OF SLAM NAVIGATION AND PATH PLANNING USING ROBOTIC PLATFORM EQUIPPED WITH LASER SCANNER AND ODOMETRY SYSTEM

пространството и могат да се ориентират автономно и да се навигират в околната среда. Това предизвиква особен интерес сред изследователите и спомага за внедряването на роботите в индустрията, селското стопанство, медицинските грижи и услугите.

Кога един робот се счита за „интелигентен“? Един от основните критерии е работата на робота в неизвестна среда. В неизвестна среда интелигентните мобилни роботи трябва да могат да изпълняват следните три задачи:

1. Картографиране. Мобилните роботи трябва да имат способността да изчертават карта на непознатата среда.
2. Локализация. Мобилният робот трябва да е наясно със своята позиция и поза в текущата среда.
3. Навигация. Мобилният робот трябва да може автономно да планира маршрут до дадено целево местоположение.

Комбинацията от задачи 1 и 2 се нарича Simultaneous Localization and Mapping (SLAM), което представлява изчислителна задача за изграждане или актуализиране на карта на неизвестна среда, като едновременно с това се следи местоположението на робота в нея.

В днешно време мобилните роботи са в състояние по зададени абсолютни координати да се придвижват до определена позиция в известна за тях среда. Най-типичният пример е използването на навигационните спътници от глобалната система за позициониране (GPS) с дадена пътна карта. Въпреки това, мобилните роботи не могат да определят своята позиция и ориентация в неизвестна среда поради ограниченията на GPS техниката:

Не може да се получи информация за околната среда. GPS може да осигури само глобално обобщена локация на базата на триангуляция, но не може да реконструира средата около устройството.

Точността на локализация достига само нивото на точност на измервателния уред. Макар и в глобален мащаб, точността на локализиране на ниво метър да може основно да отговори на потребностите, в малки по мащаб пространства точността на локализиране на ниво метър е твърде груба, за да гарантира работната точност на робота.

GPS не може да гарантира 100% точност на позициониране, а позиционирането на закрито е още по-неточно. GPS позиционирането бива повлияно от много аспекти, включително сгради, виадукти, радиовълни и т.н. Най-общо казано, колкото по-открито е пространството, толкова по-точно е позиционирането с GPS. Няма как да се гарантира, че при използване на робота, заобикалящата среда ще бъде достатъчно отворена. Невъзможността за точно локализиране на закрито директно ограничава приложението на GPS за мобилни роботи. Следователно е необходимо изследване на SLAM възможностите.

ИЗЛОЖЕНИЕ

Използване на операционна система за роботи за решение на SLAM задача

Когато роботът е в непозната среда, той трябва да знае следните три неща: къде съм? Какво е това място? Как да се махна от това място? За да се може да бъдат получени отговори на тези въпроси, роботът трябва да е въоръжен със сензори и контролери. Необходими са сензори за възприятие на околната среда. Това обикновено е камера или лазерен сензор. Роботът трябва да разполага и с метод за придвижване. Това могат да бъдат колела, крака, вериги и тн. А контролерът е това, което обработва информацията от сензорите за възприятие и управлява метода за придвижване.

Чрез сензорите за възприятие, роботът може да разбере заобикалящата го среда, така че да положи основите за изграждане на карта. Два типа сензори се използват най-често при решаването на SLAM-задачата. Единият от тях е depth камера, а другия – лазерен скенер (LiDAR).

Depth камерите имат три обектива: RGB цветна камера, използвана за заснемане на цветни изображения. Лявата и дясната страна на камерата са инфрачервен предавател и инфрачервена CMOS камера, използвана за събиране на информация за разстоянието на обектите. Разстоянието до обектите се съхранява в информациите за пикселите.

Когато роботът се движи, обектите пред сензорите за възприятие се променят, сензорите събират цялата тази информация за изображенията, която се предава към контролера за анализ и обработка. Контролерът според това изображение може да изгради карта на външната среда и в процеса на непрекъснато нарастване на картата да разбере своето местоположение (позициониране). Ако има зададени цели, роботът ще продължи планирането на пътя, за да намери най-бързия и безопасен начин за достигане до дестинацията.

Контролерът се управлява от софтуер, който кара робота да се движи, да събира информация, да избягва препятствия, да изгражда планове и т.н. Това е сложна задача, тъй като всички тези дейности са взаимосвързани (An, F., 2017). Обединяването на всички тези дейности се улеснява благодарение на наличието на ROS (операционна система за роботи). SLAM съществува под формата на пакет за разработка на ROS. Инсталиранието на този пакет позволява обработката на данните получени от сензорите за събиране на информация за околната среда, SLAM пакетът ще анализира и обработва цялостната информация и ще изгражда картата на околната среда (Buyval, A., 2017).

Най-общо казано, за да изпълни роботът SLAM задачата, той трябва непрекъснато през depth камерата или лазерния сензор да събира информация за собственото си движение и след това да анализира информацията и да изчертава собствената си позиция и околната среда върху картата. Роботът изчислява местоположението си чрез предсказване, първоначалното, то може да не е точно, тъй като входните данни са малко на брой и част от тях включват шум. Колкото повече роботът продължава да се движи, толкова повече информация събира. Когато роботът сменя позицията си, той прихваща нови характерни точки в изображението. След това роботът се опитва да свърже тези характерни точки с характерните точки, които е виждал преди. Характерните точки, които не са били виждани преди, се добавят към картата, като нови наблюдения, за да могат да бъдат повторно изследвани по-късно.

Задачата за SLAM може да бъде дефинирана чрез следните входни параметри (1,2) и очакваните резултати (3,4):

1. Управление на робота

$$u_{\{1:t\}} = \{u_1, u_2, u_3, \dots, u_t\} \quad (1)$$

2. Наблюдение

$$z_{\{1:t\}} = \{z_1, z_2, z_3, \dots, z_t\} \quad (2)$$

3. Карта на заобикалящата среда

$$m \quad (3)$$

4. Траектория на робота

$$x_{\{1:t\}} = \{x_1, x_2, x_3, \dots, x_t\} \quad (4)$$

5. Предсказването на траекторията на робота става чрез:

$$p(x_{0:t}, m | z_{1:t}, u_{1:t}) \quad (5)$$

ROS помага да се следят координатните рамки във времето. Пакетът, който се използва за това е ***tf2***. Това е библиотека за трансформиране, при която съобщенията, които се предават са от специфичен тип: ***tf/Transform*** и винаги са обвързани с една тема: ***/tf***. Съобщението ***tf/Transform*** се състои от трансформацията (транслирането и завъртането) между две координатни рамки, имена на двете рамки и времеви печат. Публикуването в ***/tf*** се извършва по различен начин от този за всяка друга тема.

Публикуване на трансформация

Създава се възел, който се абонира за ***/pose*** тема с тип съобщение ***geometry_msgs/PoseStamped*** и се публикува трансформацията между обекти, съдържащи се в ***/pose***. Необходимо е да се дефинират публикуващ обект за трансформацията ***tf::TransformBroadcaster br***, съобщение за трансформацията ***tf::Transform transform*** и кватернион за съхранение на данните за завъртане ***tf::Quaternion q***. След което се създава функция за обработка на входящите ***/PoseStamped*** съобщения, извлечане на параметрите на кватерниона от съобщението и поставянето им в структурата за кватернион, след което тя, както и параметрите за правото движение се запазват в ***transform*** съобщение, което се предава

нататък. Това става чрез `sendTransform` функцията, която има за параметър `StampedTransform`.

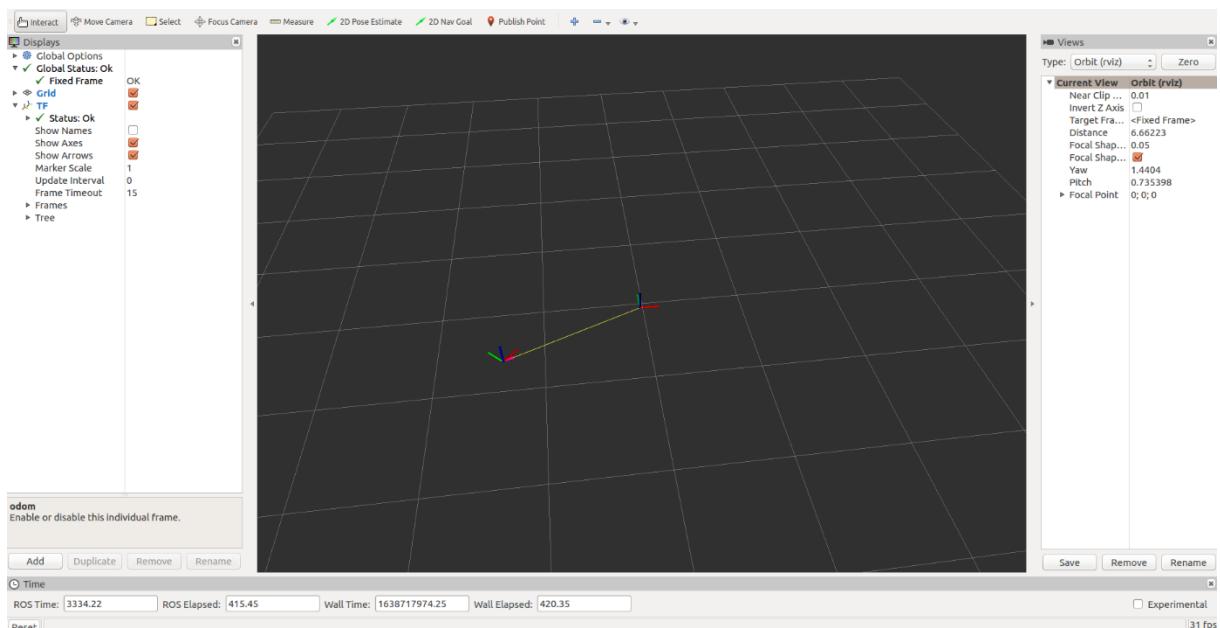
Ако е необходимо да се публикува само една статична трансформация, която не се променя във времето, достатъчно е да се използва `static_transform_publisher` от `tf` пакета. По този начин се задава връзката между шасито на робота и лазерния скенер.

```
rosrun tf static_transform_publisher 0 0 0 3.14 0 0 base_link laser 100
```

Аргументите са както следва:

- 0 0 0 – x,y,z оси на движение, стойностите са в метри
- 3.14 0 0 – z,y,x оси на въртене, стойностите са в радиани
- base_link - родителска рамка на трансформация - тази, която е статична
- laser – наследяваща рамка на трансформация - тази, която се трансформира
- 100 - забавяне между последователни съобщения

За да бъде тествано направеното дотук е необходимо да се използва контролер за управление на робота и среда за визуализация. За визуализация ще се използва `rviz`, а за управление – `teleop_twist_keyboard`. След тестване на създадения скрипт се наблюдава, че движението на рамките, които се визуализират в `rviz` отговаря на движението на робота в пространството (фиг. 1).

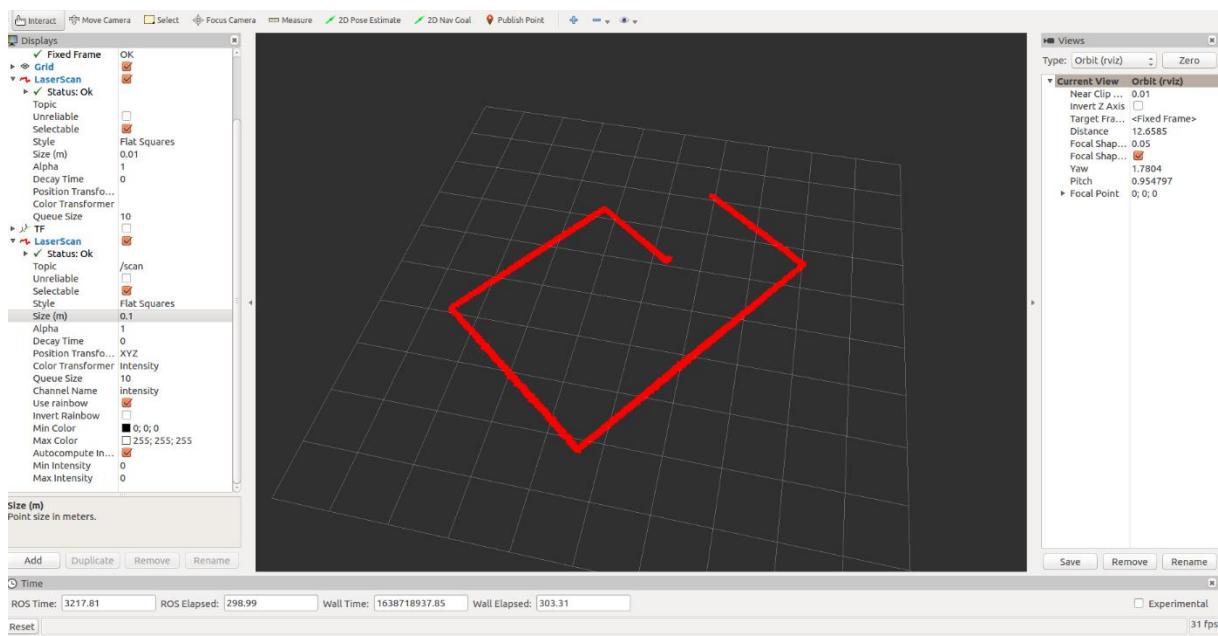


Фиг. 1. Движение на робота в пространството

За да се извърши точен и прецизен SLAM, най-добре е да се използва лазерен скенер и система за одометрия с енкодери с висока прецизност. За да се случи това е необходимо да се стартира LiDAR-а на робота, който ще публикува прихванатото от него използвайки тема `/scan` и тип на съобщението `sensor_msgs/LaserScan` (Meng, Z., 2020).

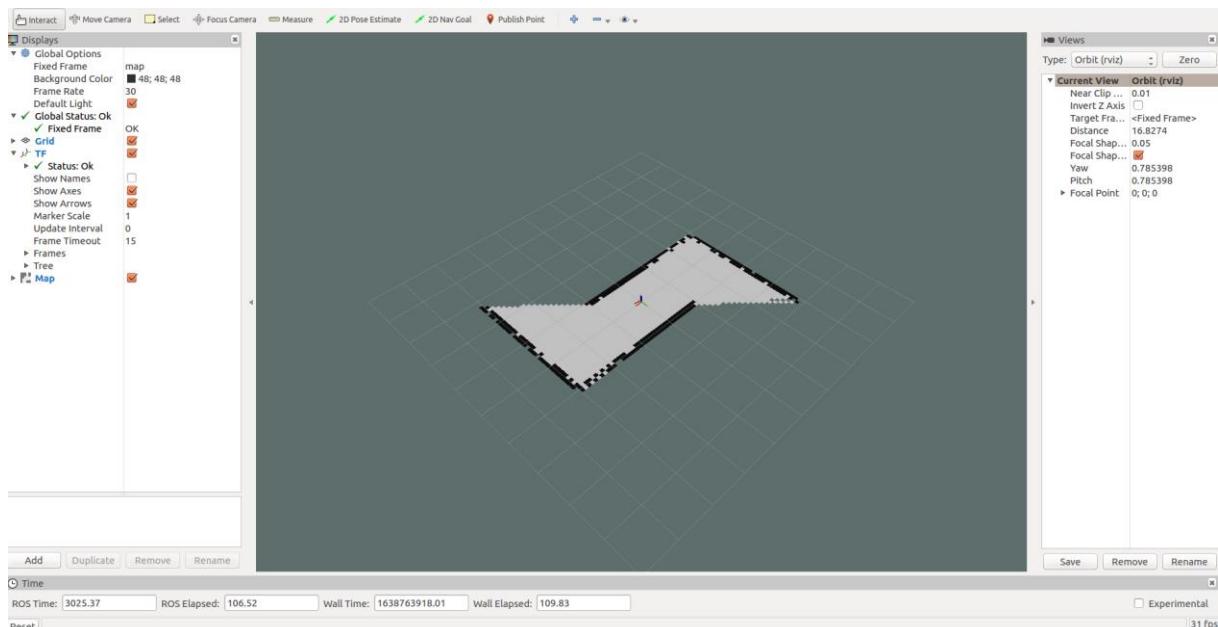
Задължително е наличието на съобщения на тема `/scan`, като това може да бъде проверено с команда `rostopic`, която позволява слушането на различни теми, но това би довело до получаването на много резултати, които са трудни за четене (Temeltas, H., 2008). Подобрият метод за проверка на темата `/scan` е да се използва `rviz`. За да се подобри видимостта на резултата от сканирането е необходимо да бъдат настроени някои от параметрите за визуализация.

След като лазерния скенер бъде стартиран, в софтуерът за визуализация вече може да се наблюдават и препятствията заобикалящи робота (Фиг. 2). Движейки робота чрез контролера, той сканира заобикалящата го среда и допълва тези граници.



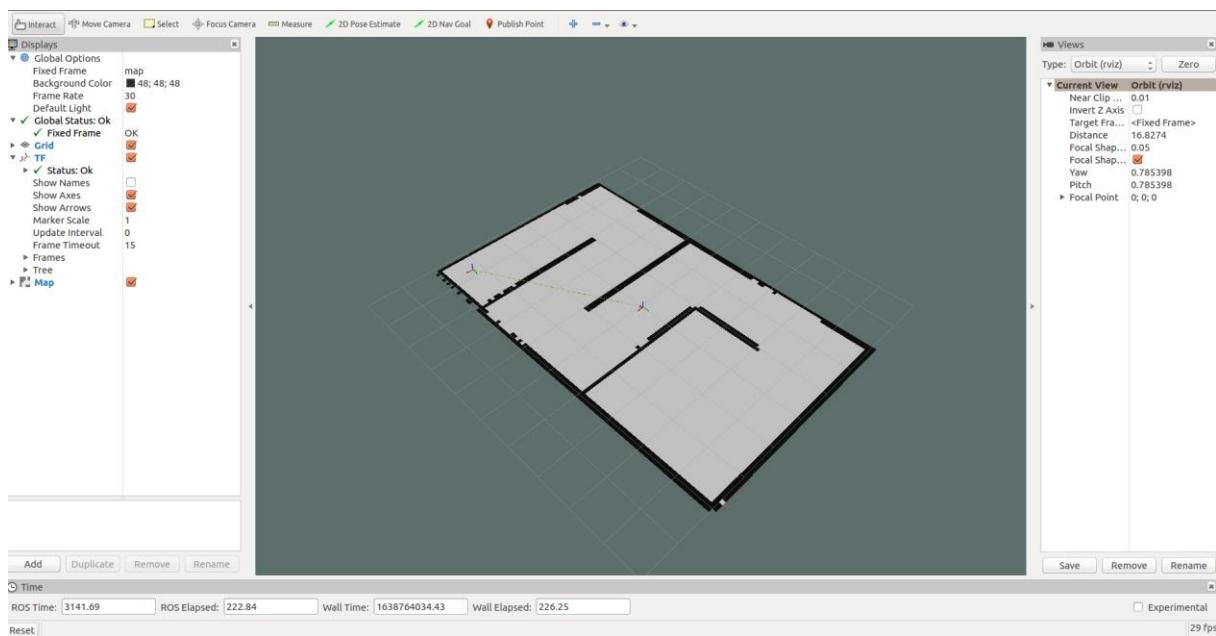
Фиг. 2. Изчертаване на препятствията

За изграждане на карта и локализиране на робота спрямо нея се използва *slam_gmapping* от *gmapping* пакета (Chen, Z., 2017). Той се абонира за */tf* тема за получаване на позата на робота спрямо начална точка и позата на лазерния скенер спрямо робота, а също така се абонира за тема */scan* за получаване на съобщенията от лазерния скенер. Също така публикува в */map* тема с тип съобщение *nav_msgs/OccupancyGrid*, което съдържа данните за картата. След извършване на необходимите настройки, при стартиране на скенера първоначално картата е почти празна (Фиг. 3).



Фиг. 3. Карта в началото на сканирането

Колкото повече роботът се движи, толкова повече информация събира и картата се запълва (Фиг. 4). Когато е обходено цялото пространство, картата става пълна и маршрутите, които се определят спрямо нея биват оптимални.



Фиг. 4 Пълна карта след изпълнение на SLAM

В тази част бяха разгледани механизмите за работа на SLAM, на пръв поглед лесното внедряване на такъв тип технология за автономна навигация бива усложнено от динамичната външна среда с променяща се светлина и движещи се обекти (Wroldsen, S. G., 2021). Справянето с такъв тип проблеми е причината да не се среща тази технология все още толкова често в човешките животи, но с нейното развитие това ще се промени.

Използване на операционната система за роботи за планиране на пътя

Планирането на пътя е основна задача за автономните мобилни роботи, която им позволява да намерят най-краткия – или казано по друг начин оптималния път между две точки. В конкретни случаи, оптималните пътища могат да бъдат пътища, които минимизират количеството завиване, количеството спиране или каквото и да е необходимо за дадено приложение (Zhang, X., 2020). Алгоритмите за намиране на най-краткия път са важни не само в роботиката, но и в мрежовото маршрутизиране, видео игрите и др.

Задачата на мобилния робот да планира пътя представлява необходимостта той да определи последователността от маневри, които трябва да бъдат предприети, за да се придвижи от начална точка до дестинация, избягвайки сблъсък с препятствия.

Примерни алгоритми за планиране на пътя са:

- Алгоритъм на Дийкстра
- A*
- D*
- Метод на изкуствено потенциално поле
- Метод на графа на видимостта

Алгоритмите за планиране на пътя могат да се основават на графи или мрежа за заетост.

Метод базиран на графи

Метод, който използва графиките, дефинира места, където роботът може да бъде и възможности за преминаване между тези места. В това представяне върховете на графа дефинират места, напр. стаи в сградата, докато дъгите определят пътища между тях, напр. врати, свързващи стаи. Освен това всяка дъга може да има присвоени тегла, представляващи трудност при преминаване на пътя, напр. ширина на вратата или необходима енергия за отварянето ѝ. Намирането на траекторията се основава на намиране на най-краткия път между два върха, докато единият от тях е текущата позиция на робота, а вторият е дестинацията.

Метод базиран на мрежата за заетост

Методът, който използва мрежата за заетост, разделя зоната на клетки (например пиксели на картата) и ги определя като заети или свободни. Една от клетките е маркирана като позицията на робота, а друга като дестинацията. Намирането на траекторията се основава на намирането на най-късата линия, която не пресича нито една от заетите клетки.

Планиране на пътя чрез мрежата за заетост използвайки ROS

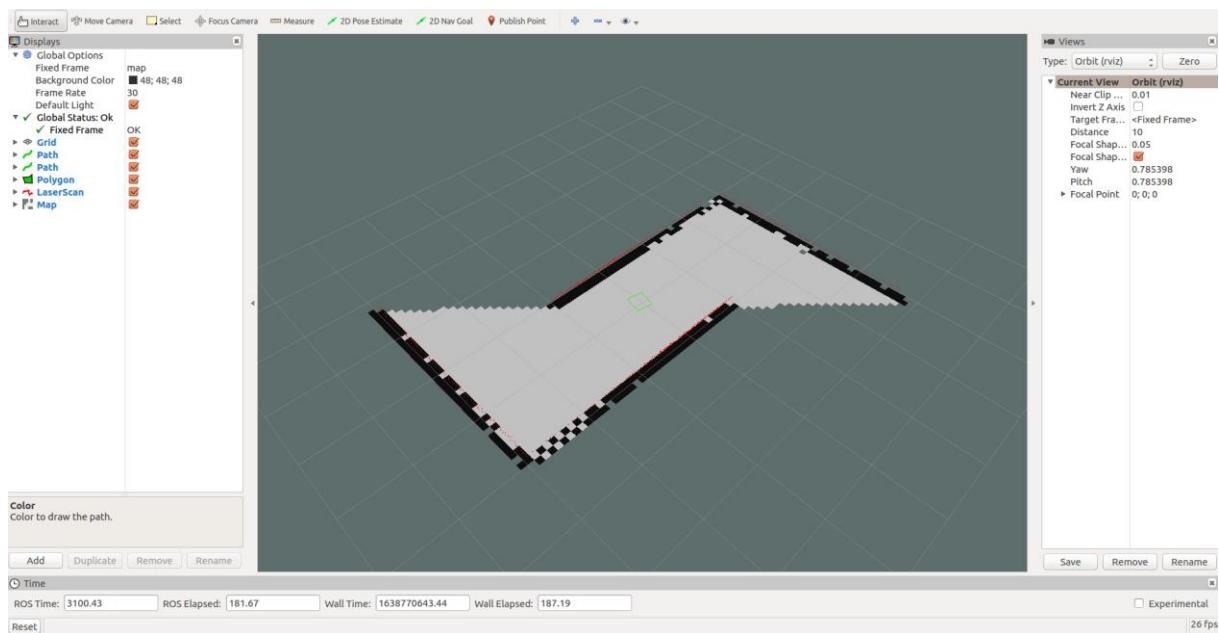
В ROS е възможно да се планира път въз основа на мрежата за заетост, напр. получената от *slam_gmapping* пакета. Планирането на пътя се извършва чрез *move_base*.

Конфигурация на *move_base*

Move_base създава карта въз основа на мрежата за заетост, която определя трудността за преминаване през определено място. Тази карта представлява мрежа, в която всяка клетка получава присвоена стойност, определяща разстоянието до препятствието, като по-високата стойност означава по-близко разстояние. Чрез тази карта се определя пътят, който е най-безпрепятствен за преминаване. *Move_base* използва две карти на разходите, локални за определяне на текущото движение и глобални за траектория с по-дълъг обхват. Локалните помагат за избягването на препятствията, докато глобалните следят достигането на крайната цел.

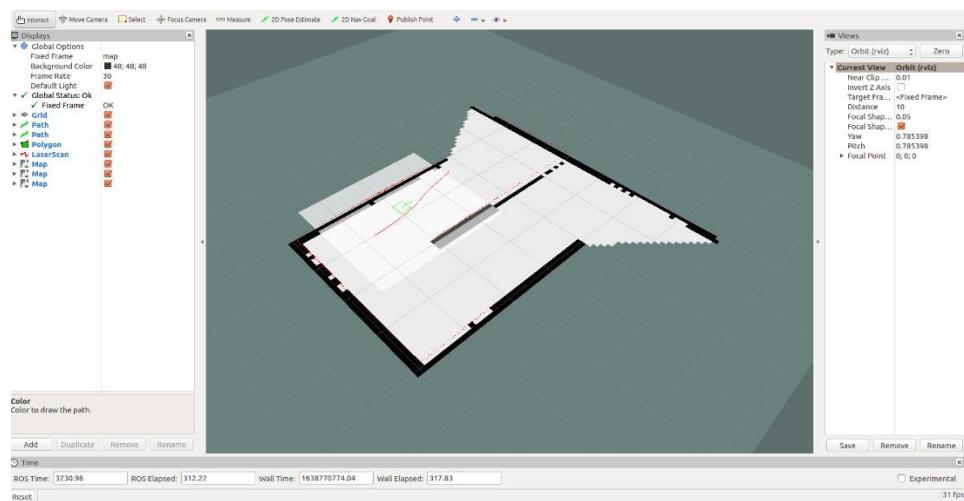
За *move_base* трябва да се дефинират някои параметри за картата на разходите и планирането на траектории, те се съхраняват в *.yaml* файлове. Техните параметри определят обхвата, в който да бъдат засичани препятствия, габаритите на робота, за да могат да се избягват сблъсъци и др. Освен това се задават параметри и за локалните и глобални карти. Чрез тях се определя, колко често да се обновява картата, колко често промените да се публикуват, дали картата е динамична или статична, размер на картата и др.

След настройване на параметрите в съответните *.yaml* файлове, скриптът е готов за изпълнение. За да се анализира резултата, отново ще се използва *rviz*. Този път ще бъдат наблюдавани освен изчертаната карта и засечените препятствия, също и локалната и глобалната карта, както и ориентацията на робота в пространството. На Фигура 5, роботът е в покой и все още няма стартирана мисия, вижда само заобикалящите го граници.



Фиг. 5 Инициализиране на планирането на път

След като всички елементи са визуализирани, следва да се зададе цел за робота. Това става като от лентата с инструменти се избере бутона 2D навигационна цел, след това се избира място в прозореца за визуализация, което ще бъде дестинация за робота. При което се наблюдава как се генерира пътят. Вижда се линия излизаша от робота сочеща посоката и роботът се движи към местоназначението си (Фиг. 6).



Фиг. 6 Движение на робота към местоназначението

ЗАКЛЮЧЕНИЕ

По настоящем развитието на алгоритмите за планиране на пътя прогресивно се ускорява поради намирането на нови приложения като дизайн на платки, мрежови маршрути, компютърни анимации и т.н. Изследователската общност предлага много подходи за решаване на проблема с планирането на пътя. Тази статия разглежда подходите за планиране на пътя и решаването на SLAM задачата при мобилните роботи. От тестовете се наблюдава, че мобилният робот се справя без проблем в лабораторни условия, а постоянно подобряване на алгоритмите и сензорите ще направи тези технологии все по-често срещани в ежедневието.

БЛАГОДАРНОСТИ

Тази публикация е подпомогната от Националната научна програма „Информационни и комуникационни технологии за единен цифров пазар в науката, образоването и сигурността (ИКТвНОС)”, финансирана от Министерството на Образованието и Науката на Република България. Публикацията отразява част от резултатите, които са получени в рамките на Проект BG05M2OP001-1.001-0004 - Университети за Нauка, Информатика и Технологии в e--обществото (УНИТe), финансиран по Оперативна програма „Наука и образование за интелигентен растеж“.

REFERENCES

- An, F., Chen, Q., Zha, Y., & Tao, W. (2017, October). Mobile robot designed with autonomous navigation system. In Journal of Physics: Conference Series (Vol. 910, No. 1, p. 012036). IOP Publishing.
- Buyval, A., Afanasyev, I., & Magid, E. (2017, March). Comparative analysis of ROS-based monocular SLAM methods for indoor navigation. In Ninth International Conference on Machine Vision (ICMV 2016) (Vol. 10341, p. 103411K). International Society for Optics and Photonics.
- Chen, Z., Weihua, S. U., Weining, A. N., & Qin, X. (2017). SLAM and path planning of mobile robot in ROS framework. Chinese Medical Equipment Journal, 38(2), 109-113.
- Meng, Z., Wang, C., Han, Z., & Ma, Z. (2020, September). Research on SLAM navigation of wheeled mobile robot based on ROS. In 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE) (pp. 110-116). IEEE.
- Temeltas, H., & Kayak, D. (2008). SLAM for robot navigation. IEEE Aerospace and Electronic Systems Magazine, 23(12), 16-19.
- Wroldsen, S. G. (2021). LiDAR-Inertial SLAM System for Tunnel Navigation of an Autonomous Road Roller (Master's thesis, NTNU).
- Zhang, X., Lai, J., Xu, D., Li, H., & Fu, M. (2020). 2D LiDAR-based SLAM and path planning for indoor rescue using mobile robots. Journal of Advanced Transportation, 2020.