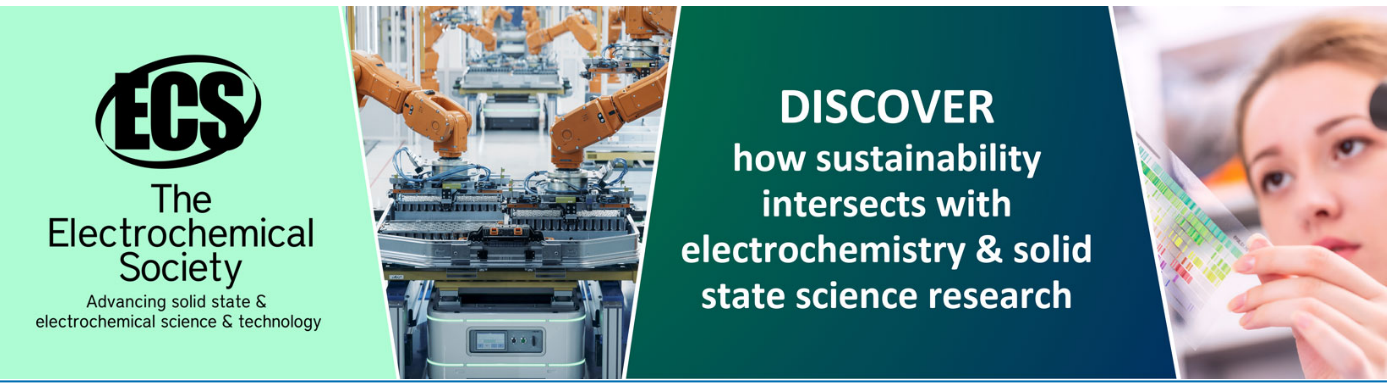
IOP Conference Series: Materials Science and Engineering

PAPER • **OPEN ACCESS**

Bricklaying robot moving algorithms at a construction site

To cite this article: A V Malakhov *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **734** 012126



IOP Conf. Series: Materials Science and Engineering **734** (2020) 012126 doi:10.1088/1757-899X/734/1/012126

Bricklaying robot moving algorithms at a construction site

A V Malakhov[1], D V Shutin[2] and S G Popov[2]

[1]Department of Unique Buildings and Structures, Southwest State University, 94 50 Let Oktyabrya street, Kursk, 305040, Russian Federation
[2]Department of Mechatronics, Mechanics and Robotics, Orel State University named after I.S. Turgenev, 95 Komsomolskaya street, Orel, 302000, Russian Federation

E-mail: 4ega@inbox.ru

Abstract. Automation of bricklaying using mobile robots requires new solutions in many spheres, including developing new algorithms. A mobile bricklaying robot moving at a construction site is considered as a task of planning its trajectory. The task is partly similar to passing a maze but has some distinctive features. Analysis of required movements and analysis of structure of a typical masonry plan allows create an approach to building a moving algorithm for a bricklaying robot. The developed algorithms consist of two separate algorithms, one for moving while directly building walls, another for fast and effective travelling between two points of a map without laying bricks. The developed algorithm has to be tested for different masonry plans and may be optimized in the described ways.

1. Introduction

Automation of bricklaying is usually implemented using special robotic systems. Principles of operation of such systems may be different. Stationary robotic systems are mechanically attached to the coordinate system of a building being constructed. Such approach is connected with limited size of the system and, consequently, limited size of the building. An example of such robot is Hadrian [1]. It is fixed before beginning of construction and takes this position during all working time, only manipulator moves. Mobile robots do not have a rigid connection with the coordinate system of the constructed object. Due to this fact the task of positioning of the working instrument of a robot becomes more difficult. With that, mobility of such robots allows build constructions with almost unlimited size. Examples of such robots are SAM100 [2], and the robot that will be considered in this article [3, 4]. The robot contains the 4-degrees of freedom manipulator mounted on a moveable chassis. The manipulator takes a block from a bricks supplying device and moves it to its place in a wall. Blocks are laid layer by layer, layers the layers correspond to the rows on a masonry plan.

On of main problems for mobile bricklaying robots is creating an effective algorithm of moving at the construction site. Such algorithm must be universal enough due to the fact that most buildings in projects for such robots differ from each other. The result of the algorithm's operation is a plan of robot movements at the construction site during laying all bricks according to a masonry plan.

2. An overview of algorithms

The task for a bricklaying robot that should be solved by the considered algorithm is in some aspects similar to a well-known problem of searching an exit in a maze. This is due to the fact that a masonry plan usually has a structure similar to a typical maze, branched and having many ways and turns divided by walls. There are several known algorithms widely used for solving such problem: Flood Fill algorithm [5], a Wall Follower [6, 7], more specific and improved algorithms like [8, 9]. Some papers provide a greater range of algorithms for a maze [10]. However, such algorithm cannot be applied to the considered task. For example, a Flood Fill algorithm does not make robot tracking walls for laying blocks, it is only searching for and exit. A WallFollower algorithm operates only with an assumption that all walls a connected to each other, and separately situated walls are skipped, so not all walls will be built. So, a special algorithm should be introduced in order to take into account all the features of the considered task for a bricklaying robot.

Firstly, a more detailed analysis of the task should be held. Analysis of all robot movements that should be made at the construction site allows dividing them into two categories:
1) movements during laying bricks – such movements are implemented directly along walls being built so that these walls would have been within the manipulator's operation zone;
2) free movements during which blocks are not laid – such movements can be implemented the following situations:

- when a robot moves past the section of a wall that has been already filled with necessary blocks towards a remote unfilled section;
- when a robot moves from one wall section to another that are not connected to each other within a layer being built;
- when a robot moves to the start point.

The first category of movements is mostly determined by a required amount of walls to be built so it is difficultly optimized. It takes into account the variety of possible built architectural forms and methods of implementing them and generates a trajectory of passing the brickwork (all length of all walls). Such algorithm will be called a "Brickwork Passing Algorithm" (BPA).

On the contrary, free movements should and can be optimized because there are different combinations of such movements and extra amount of free movements reduces the economic efficiency of a bricklaying robot that is based on its high productivity. So it worth it to use special methods and algorithms of searching the less power-consuming combinations of algorithms. Such algorithm will be called a "Free Movements Algorithm" (FMA).

The basic description of such algorithms is the topic of the present paper and is shown below.

3. A brickwork passing algorithm (BPA)

The considered mobile bricklaying robot is equipped with rectangle shaped manipulator's working area. The local coordinate system of a manipulator consists of axes X1, Y1, Z1, while the global coordinate system of a building includes axes X, Y, Z. The scheme of axes is shown in figure 1. The working zone of a robot is at a certain distance from its chassis, so there is a minimal working distance d_{min} from a chassis to a wall being built. Increasing the distance leads to reducing the working area of a manipulator, decreasing it leads to impossibility of laying the closest bricks. So, the distance should be kept closest to a d_{min} value during all the robot operation according to the BPA. This distance also determines the parameters of trajectory control during robot's movements.

Let's describe the general features of the BPA. Robot is able to put blocks at their sites in masonry when such sites can be covered by projection of a manipulator's working area. So, BPA should generate a trajectory that should allow covering all walls projection with the projection of a manipulator's working area. So, a shape of walls projection at a building plane determines the robot's trajectory being calculated with the BPA.

The common features of shapes of walls at a masonry plan from the point of view of a robot movement algorithm are as follows:

- walls are oriented orthogonally relative to the base plane (plane of a construction site, particularly floors);

- walls in vertical projection are one or combination of several objects with branched and extended shape;
- projections of each row of blocks to the basic plane can be considered as same, and not taking into account for making the movement algorithm the local voids, like window and door openings;
- a wall thickness can be different at different parts of a building, at that walls of thickness more than three full block sizes are rare in practice (this fact also determines the size of the operating zone of a robot's manipulator at Y1 axis);
- a projection of a wall to a basic plane is a contoured object consisting of orthogonal segments of various length, and robot during bricklaying process follows along such segments; with that, the closest to the chassis point of manipulator should always be outside the contour within the admissible deviation distance;
- a surface of a wall that robot travels along is a sequence of orthogonal segments of various length, and a robot's trajectory is also aligned by the segments;
- it is also necessary to take into account a mutual orientation of segments within one wall that shows its thickness and structure.

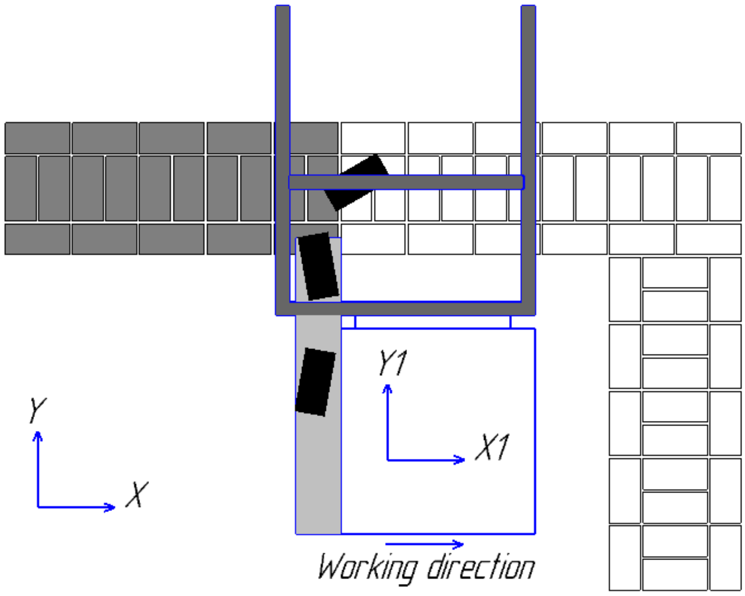


Figure 1. Coordinate systems.

A robot's trajectory should be a cyclical pass along all walls' length. The number of cycles should be equal to a number of rows in the masonry. In order to optimize the trajectory and reduce the bricklaying time, a robot should visit each point of a wall minimum times, at least one.

So, calculating a trajectory of robot moving at a construction site consists of the following stages:

- analysis of size and configuration of the building being constructed for one layer;
- making a sequence of structures to be passed, with start and finish at one point;
- setting a number of iterations of passing such trajectory equal to a number of rows of blocks.

Since there are typical structures in walls configuration, BPA searches them among shapes of walls. So, all wall contours that should be passed by a robot during bricklaying may be considered as a sequence of such typical structures with known algorithms of processing, and only parameters should be adjusted. Such typical shapes are illustrated in the figure 2.

Let's consider the typical structures and ways of processing them by a robot according to the BPA. The processing includes moving along a structure and putting all blocks to its sites according to the masonry plan.

The basic typical structure is a rectilinear section. According to BPA a robot follows along such section at a constant distance of d_{min}.

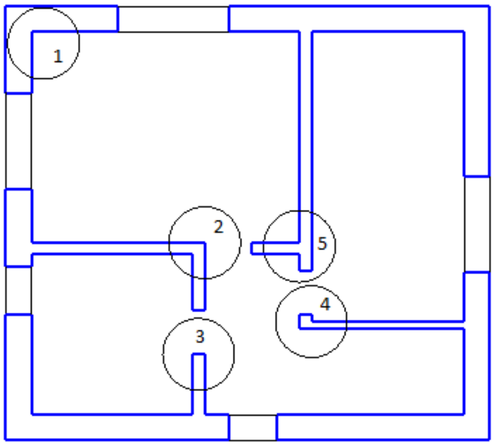


Figure 2. Typical structures to be processed by a bricklaying robot
(1 – an internal corner, 2 – an external corner, 3 – a wall end, 4 – a protruding element, 5 – a junction).

External and internal corners are simple non-linear structures (positions 1 and 2 in the figure 2). Robot makes external and internal turns in such places. It turns 90 degrees in both cases, but with different movements and start and stop points of such turns. The sequence of a robot's movements is shown in the figure 3, a) and b). In order to provide maximal coverage of a wall with a working area of a manipulator, passing an internal corner requires making a return linear movement before rotation in place. An external corner is passed along a curved path around a corner point.

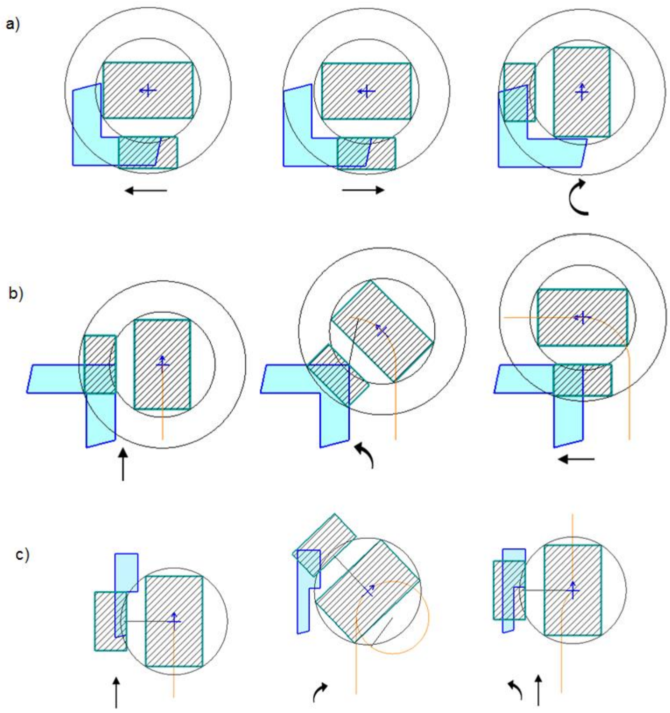


Figure 3. Typical maneuvers of BPA.

Passing a wall end can be considered as a couple of external turns with or without a linear segment between them, depending on a wall thickness.

Passing junctions is also considered as passing internal and external corners and, in some cases, protruding elements that are described below.

A protruding element is an element (e.g. a pilaster) that protrudes beyond a wall line not more than a depth of a manipulator's working zone h, otherwise it is considered as a junction. A protruding element is passed by a robot with preliminary moving away from a wall (figure 3c).

It also should be noted that BPA operates with concepts such as closed and open bends. An open bend is a bend of walls contour that is an orthogonal obstacle for a robot moving towards it (e.g. when moving towards an internal corner). A closed bend does not provide an obstacle for a robot (e.g. when it moves towards an external corner). An open and a closed bends are called opposite.

The general description of BPA is as follows.

1.Choosing a start point in a center of a linear segment with length of at least two times greater than a maximal robot dimension.

2.Start following the contour to a robot's working direction.

3.Detection the type of a first found non-linear structure:

a. if first bend is followed by an opposite bend, and the length of a linear segment between them is less than h, the found structure is a protruding element, at that in case of the first bend is closed, the structure is a deepening, otherwise it is a protrusion;

b. if a first bend is followed by a same bend, the found structure is a turn, at that in case of the first bend is closed, it is an internal turn, otherwise it is external.

4.The sequence of the required robot maneuvers is saved in a memory:

- linear moving from the start point to the first structure (the point at which a maneuver should begin);
- the sequence of action making up a maneuver.

5.If the upcoming linear section includes the start point, finish implementing the algorithm, otherwise repeat from the p.2.

4. A free movements algorithm (FMA)

Free movements do not require following the contours of walls and should take the shortest time to be passed. As it has been mentioned above, the FMA should be based on algorithms allowing searching the shortest (and so less power- and time-consuming) ways between a start and a final point. So, the LPA* [11] has been chosen as the basic algorithm for this task as it meets the mentioned requirements and also allows avoiding unexpected obstacles that can take place at a construction site.

The map for the FMA algorithm is also based on a masonry plan. Walls are considered as persistent obstacles, at that presence of all blocks in all layers are logically add up. The result is a list of points describing the free areas and areas with obstacles. Additionally, the description of each point also contains its coordinates (X, Y), value of energy intensity (equal for all free points due to the flatness of a construction site), data about presence of an obstacle in the point, data if the point has already been visited and if it belongs to the generated path.

The path is generated according to the rules of standard LPA* algorithm. Unexpected obstacles are determined by a robot during passing the way using proximity sensors on board. Once a new obstacle has been detected, it is added to the map of the FMA algorithm and the path is calculated again.

A distinctive feature of the FMA algorithm is a priority queue implemented using the dynamically specified memory. The priority queue algorithm itself is not optimal as each new added element to the queue leads to new sorting of the queue, that is implemented using merge sort.

5. The overall movement algorithm of a bricklaying robot

Combination the BPA and the FMA gives the overall algorithm for planning movements of a mobile bricklaying robot at the construction site. It is as follows:

1. Preliminary calculation movements along the walls with the BPA based of a masonry plan and loading the result to the memory of a robot.
2. Determining the start point of a robot and the current position at the map.
3. Building a path to the start point with the FMA and moving to it.
4. Beginning bricklaying with moving along the trajectories calculated with the BPA.
5. If a zone with already laid bricks is reached:
5.1. If not all blocks at the layer are put:
   a. the upcoming area with not put blocks is determined;
   b. the shortest way to the determined point is calculated with the FMA and moving to it;
   c. continuing operation at p.4.
5.2. If not all blocks at the layer are put, operation at the layer is finished, and laying the next layer begins at p.3.
6. If the last (highest) layer is finished, stop operation of a robot.

6. Conclusion

Development of a mobile bricklaying robot includes studying typical brickwork parameters in order to create effective algorithms of making them by automation means. From the point of view of automatics, typical walls made if bricks have some typical features. Knowing these features allows reducing the required range of robot functionality and simplify its algorithms. The found features and assumptions were used when developing a complex algorithm of moving a mobile bricklaying robot at a construction site. This algorithm is based on a combination of two separate developed algorithms, the Brickwork Passing Algorithm (BPA) that is used during the direct making a brickwork, and the Free Movements Algorithm (FMA) for optimal moving robot at a construction site without laying blocks.

The further improvements of the complex bricklaying robot moving algorithm will be made in the following directions:

- increasing universality, including processing more complex structures of buildings;
- increasing the overall efficiency, firstly by further optimization of mechanism of generation robot routes at vast and highly branched objects;
- increasing the performance of implementing an algorithm by using improved math, applying improved algorithms for the required tasks, including taking into account properties of used software and hardware.

References

[1] Robogeek.ru 2019 URL: *http://www.robogeek.ru/promyshlennye-roboty/robot-adrian-mozhet-postroit-dom-za-dva-dnya*
[2] Green-city.su 2019 URL: *http://green-city.su/rukami-robota/*
[3] Malakhov A V and Shutin D V 2019 The Analysis of Factors Influencing on Efficiency of Applying Mobile Bricklaying Robots and Tools for Such Analysis *J. of Phys.: Conf. Ser.*
[4] Malakhov A V and Shutin D V 2019 Applying the Automated and Robotic Means for Increasing Effectiveness of Construction Projects *Trans Tech Period.:Mater. Sc. Forum*
[5] Burtsev S and Kuzmin Ye P 1993 An Efficient Flood-Filling Algorithm *Comp. & Graph* **17** 549-61
[6] Fahmizal F, Arrofiq M and Mayub A 2018 Identifikasi Pemodelan Matematis Robot Wall Following *Jurnal Nasional Teknik Elektrodan Teknologi Informasi* **7**
[7] Garcia M, Rodriguez A, Munoz J and Rangel P 2019 Maze Solver Algorithms Inspired on Micromouse Competition *Couri Symposium Spring 2019*
[8] Tjiharjadi S, Wijaya M and Setiawan E 2017 Optimization Maze Robot Using A* and Flood Fill Algorithm *Int. J. of Mech. Eng. and Rob. Res.* **6** 366-72
[9] Musridho R, Yanto F, Haron H and Hasan H 2018 Improved Line Maze Solving Algorithm for

Curved and Zig-zag Track *Proc. of 7th ICT Int. Stud. Proj. Conf.*
[10] Niemczyk R and Zawiślak S 2020 Review of Maze Solving Algorithms for 2D Maze and Their Visualisation *Container Handling in Automated Yard Blocks* (Springer International Publishing) pp 239-52
[11] Likhachev M and Koenig S 2004 Lifelong Planning A* *Artif. Intel.* **155 (1-2)** 93-146
[12] Siciliano B, Sciavicco L, Villani L and Oriolo G 2009 *Robotics. Modelling, Planning and Control. Introduction* (London: Springer-Verlag) p 630