On the Bar Installation Order for the Automated Fabrication of Rebar Cages [*]

Mahdi Momeni[1], Johan Relefors[2], Lars Pettersson[3], Alessandro V. Papadopoulos[1], and Thomas Nolte[1]

[1]Mälardalen University, Västerås, Sweden
[2]Solving Robotics Sweden AB, Västerås, Sweden
[3]Skanska Sweden AB, Stockholm, Sweden

mahdi.momeni@mdu.se

Abstract -

Robotics automation is a promising solution for the fabrication of structures made out of reinforced concrete. The reinforcement is often installed directly in the form and bar-by-bar. Using bigger pre-fabricated units (cages) may be beneficial for saving construction time and better labor safety. In this paper, we focus on the problem of automating the generation of a plan for the installation of rebars, given the digital twin of a desired reinforcement cage design, and of its basic components. More specifically, the plan describes the assembling order for the rebars such that (i) it is possible to fabricate the reinforcement cage by the robots, and (ii) the end product is the final reinforcement cage, ready for installation in the form for the concrete structure. In this paper, we propose an algorithm to automatically compute a feasible installation order for a generic rebar cage. The feasibility of the generated order is also case studied and simulated on a simplified rebar cage under the given assumptions.

Keywords -

Robotics for construction; Rebar installation; Automation

1 Introduction

The process of installing reinforcement bar-by-bar in reinforced concrete structures is time-consuming and potentially harmful for the operators' safety. Pre-fabricating the reinforcement in bigger units (often referred to as "cages") and placing these units in the form may help save construction time. This paper focuses on the usage of industrial robots for the automated construction of rebar cages, aimed to minimize the impact on the operators' safety and to optimize the construction time. Automatic fabrication of rebar cages can save the construction time even further, among other benefits and challenges which can be potentially added.

One of the major challenges in automating rebar cage fabrication is that they are high mix and low volume. The automation cannot rely on repeating the same motions over and over, but a higher degree of flexibility is required. This challenge can be divided into two major challenges. First, the adopted industrial robots require accurate motion planning that cannot be easily reused between one cage and another. Second, the ordering sequence of one cage can significantly differ from another one. This calls for novel algorithmic approaches that can handle such flexibility.

Figure 1. Gantry robot system.

In [1], we explored how the first challenge can be addressed, by automatically generating motions for a gantry-robot setup with 27 degrees-of-freedom (DOFs), shown in Fig. 1. However, the installation order was tailored to, and manually determined for the specific cage used in the experimental demonstration, therefore limiting the possibility of reusing the same solution for different types of cages.

In this paper, we focus on the problem of identifying a feasible installation order, given a generic rebar cage. With "installation order" we mean the determination of how and in what sequence the rebars should be placed to assemble the cage. If no such *valid order* exists, or if one cannot be found, the cage is considered not *robotically fabricable*. A valid order should take into consideration the gripping poses and tying poses while ensuring that collision-free paths for placing the rebars exist. Moreover, it should ensure that the rebar cage remains stable during fabrication and at the end of the process. Note that many valid installation orders may exist but in this context finding one such order is enough.

---

[*] This work was supported by the Knowledge Foundation with the Automation Region Research Academy (ARRAY) and SBUF.

2 Background

In [2], the authors used Constraint Satisfaction Problem (CSP) to formulate sequence and motion planning problems (SAMP) for spatial extrusion of 3D trusses. The authors present an automated approach to finding an extrusion sequence. In their work, they refer to sequence as the order of the end-effector's feasible direction for each extrusion direction. In this paper, however, the sequence refers to the order following which the rebars will be placed one after another.

In [3], the authors present an automated planning approach to find a construction sequence and plan robot motion jointly for additive manufacturing. In their work, however, they define a construction sequence as an ordering of the elements which are welded or glued together at their ending. On the other hand, they are only dealing with straight elements. In this work, the considered rebars can be connected at any point on the bar and the rebars can have different shapes.

In [4], the authors presented a fabrication process method based on a cooperative assembly approach. They use two robots to cooperatively fabricate a full-scale vault brick by brick. They design the construction sequence taking into account the temporary stability of neighboring brick assembly while maintaining global stability. They develop their approach based on the principle that the added self-weight should be efficiently transferred to the foundation. They are also working with identical bricks. In our work, the rebar cage is built using a gantry robot system (shown in Fig. 1). In addition to that, a rebar cage in our problem can contain hundreds of rebars, many of them of different types.

The problem of finding the bar installation order is split into two sub-problems: (i) finding a way to deal with a large number of ordering possibilities, which is a combinatorial problem, and (ii) finding gripping and tying poses, as well as the robots' trajectories. More specifically, we analyze a rebar cage structure to reduce the complexity of the problem by reducing the search space. To this end, we order the rebars based on the geometry of the rebars and the way that the path planning algorithm presented in [1] works. We test this ordering on a simplified cage. The input used is a digital twin of the cage, i.e., a digital version of the cage where all necessary bars are present. More specifically, the digital twin contains the position and geometry of all the bars in the cage. Each bar has a unique ID. This ID is defined using information such as: (i) the bar mark which contains information regarding the bar shape type and a serial number which identifies a group of identical bars; (ii) a serial number identifying an individual bar in a group of identical bars; and (iii) the way the bar should be fixed to other bars, e.g., welding or tying.

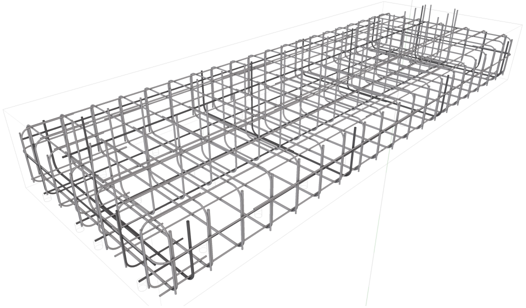

Figure 2. The digital twin of a cage.

The remainder of this paper is organized as follows. Section 3 presents the problem addressed in this paper. Section 4 describes the methodology developed to compute the bar installation order. Section 5 discusses the obtained results, and Section 6 concludes the paper.

3 Problem Statement

The problem of finding a rebar placement sequence from a digital twin can be formulated as follows:
Given the digital twin of a rebar cage and an automation system, find a feasible order in which the bars can be installed by the system, one after another, to fabricate the entire rebar cage.

Finding such a sequence is far from being trivial, due to its combinatorial nature [5]. The main focus of this work is to investigate ways of reducing such complexity by exploiting the nature of the problem and the domain knowledge and expertise. In particular, in this paper, we focus on finding an installation order for a single cage while using rebar trajectories that are compliant with the trajectories of the path planning algorithm proposed in [1].

We are then looking for an installation order for a cage with the following constraints and assumptions:

1. The rebars are placed following translational movements complying with the path planning algorithm in [1].
2. The rebars should remain connected to each other at each installation step[1]The rebar should remain connected to, at least, one other rebar(s) from either the same or another layer(s) at each installation step[2], except for the rebar(s) in the first layer which is assumed to rest on the ground and may not connect to another rebar(s).
3. The rebars are stiff, i.e., their geometry corresponds to the geometry given by the digital twin.

Theoretically, the number of installation orders is equal

---

[1] No rebars are allowed to float in mid-air

[2] No rebars are allowed to float in mid-air

Algorithm 1: Forward Method Algorithm.

```
input  : $DT$ = Digital Twin
output : $R$ = Placement Sequence
$Rebar$ ←RebarData($DT$) ;    /* Determines
    rebars parameters: ID, type, and
    position w.r.t the system coordinate
    framel */
[$s, Z$] ←Layers($Rebar$) ;  /* s: Number of
    layers, Z: HLs coordinates, each layer
    includes HLs with the same
    $z$−coordinates */
for $j$ ← 1 to $s$ do
    // find all the rebars with at least
       one Horizontal Leg in layer $j$
    $tmp$ ←RebarsInEachLayer($Z(j)$);
    // sort rebars based on their “x-”,
       “y-” and “z-”coordinates
    $SortedRebars(j)$ ←SortRebars($tmp$) ;
    /* from left to right */
end
$R$ ← $SortedRebars$
```

to the factorial of the number of rebars in each cage. Solving the problem by enumerating the solutions is therefore not possible. However, as we are interested in finding one feasible solution, we can stop evaluating orders as soon as a solution is found. This suggests using a backtracking and Depth-First Search (DFS) [6] algorithm to solve the problem.

To further analyze the structure of the problem, we look at all states that the cage can assume during fabrication, denoting this as the *states of a cage*. A key observation is that the same state of the cage can be part of different installation orders. The history of how the cage came to a state is not important. Saving the computations that follow from a certain state of the cage means that the computations following from that state do not have to be performed again if another order leads to the same state of the cage.

Summarizing the analysis, the problem is well suited for a depth-first search algorithm. Furthermore, saving any computations made during the search to re-use them is important since the same state of a cage can be visited more than once. Finally, the criteria of complying with the path planning algorithm mean that the depth-first search should be guided in a direction where blocking yet-to-be-placed rebars are avoided.

4 Methodology

In this section, we present a heuristic to compute the bar installation order that we call the *Forward Method*, presented in Algorithm 1. In this method, rebars are added one by one starting from scratch. This can be compared with the *Backward Method*, presented in [7], which is based on the main principle of picking the cage apart, bar by bar.

(a) A-Bar (b) B-Bar (c) C-Bar

Figure 3. Different rebar types used in this paper.

As motivated in the previous section we are going to use a depth-first search to find an installation order. The search needs to be guided in a way that increases the likelihood of finding a feasible installation order. This means that the path planning algorithm has to be taken into account.

The planning algorithm is designed to place rebars by moving them straight down, followed by an approach movement. Looking at a cage from above, the horizontal rebar legs have a larger surface area than the vertical rebar legs. We interpret this to mean that horizontal legs in a partially assembled cage are more likely to block the installation of further rebars. This is especially true for horizontal legs which are at the top of the cage.

To use this idea to guide the search algorithm we assign a height coordinate to each rebar, denoted as the Z-coordinate of the rebar, as the height of the top horizontal leg of the rebar. The height is defined with respect to the system coordinate frame which can be arbitrarily placed anywhere on the floor. We then initialize the search algorithm with a list of rebars where the order is from lower to higher Z-coordinate. The list is then used in the search by adding rebars to the cage in the order in which they show up in the list.

There may be, and often is, many rebars with the same Z-coordinate in a cage. We denote collections of rebars with the same Z-coordinate as being in the same *layer*. In a given layer the order in which the rebars should be placed is not guided by the Z-coordinate. This freedom is used by ordering the rebars in a given layer by their distance to the origin of the system coordinate.

The inclusion of an arbitrary origin may seem strange at first. The rationale is that starting from some point in space should maximize the size of free space around the rebars in a partially constructed cage. Note, however, that changing the position of the origin can affect both the installation order and the convergence of the search.

5 Results and Discussion

The pseudo-code of the Forward Method algorithm is explained in Algorithm 1. We used MATLAB®[3] to implement the algorithm. To test the algorithm, we have chosen

---

[3] www.mathworks.com

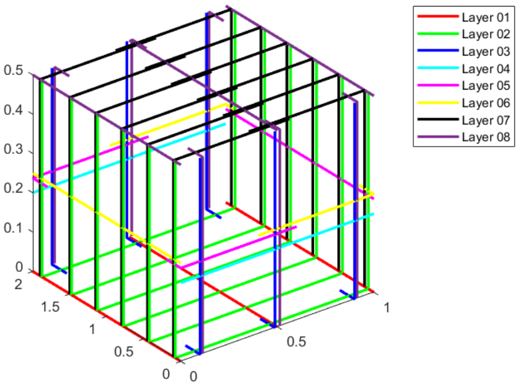

Figure 4. The simplified model of the cage in Fig. 2.

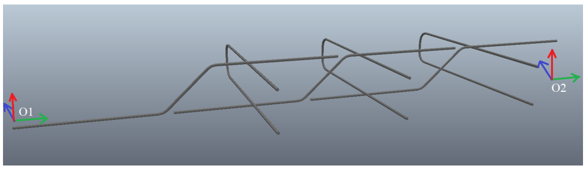

Figure 5. A row of 6 bars of two different types.

the digital twin of a reinforcement cage, see Fig. 2, which was designed for a bridge structure in Stockholm, Sweden. For the sake of this paper, we simplified the cage to the one with fewer rebars (from 198 bars in the original one down to 44 bars), see Fig. 4. Noting that there might be loss of generality with any simplification, this simplification is done for the digital twin to be imported to MATLAB® for a faster prototyping phase. For the final product, however, the software we are working on is being developed in C++ and will take the original digital twin directly as the input.

The simplified cage contains three types of bars, shown in Fig. 3. For this case study, the algorithm can find a sequence without backtracking in the depth-first search. An order is also produced regardless of the position of the coordinate frame[4]. There are overall 8 different layers in this case study which means that the cage will be fabricated in 8 different stages. The number of translational movements that the robots need to make to fabricate the cage, however, is equal to the number of rebars in a cage. Different layers are shown in different colors in Fig. 4.

Unlike the presented case study, we can find cases where the position of the origin affects the number of branches that need to be explored in the search. An example is given by the collection of rebars shown in Fig. 5 which contains 6 rebars. Depending on whether the origin is set to $O1$ or $O2$ the number of branches that need to be explored and backtracking differs.

6 Conclusion

Finding a valid installation order so that the robots can automatically fabricate a reinforcement rebar cage is a challenging task. Especially if we take into account the gantry robot's trajectory to install the rebars as well as all the many different rebar types. We, therefore, tried to split the problem into two subproblems of (i) finding a sequence(s) of rebar placement as an initial starting point, and (ii) verifying that the determined sequences are executable by gantry robot systems. We addressed the first part of the problem in this paper and presented an algorithm to find a sequence(s) for placing the rebars in a cage.

Theoretically, the problem of finding the bar installation order is combinatorial with respect to the number of rebars in the cage. This paper proposes a heuristic to be able to compute a feasible solution. The presented algorithm requires the rebar cage to be connected in each step of the assembly process. This is a relaxation of the requirement that the cage should be stable in each step. This relaxation needs to be addressed in future work.

We also need to address the second part of the problem. One idea in that direction is to use the algorithm presented in this work to produce several different installation orders, possibly by moving the origin into different locations, and see if any of these orderings is fully compatible with the path planning used for the gantry-robot system.

References

[1] M. Momeni, J. Relefors, A. Khatry, L. Pettersson, A.V. Papadopoulos, and T. Nolte. Automated fabrication of reinforcement cages using a robotized production cell. *Automation in Construction*, 133:103990, 2022.

[2] Y. Huang, C.R. Garrett, and C.T. Mueller. Automated sequence and motion planning for robotic spatial extrusion of 3D trusses. *Contr. Rob.*, 2:15–39, 2018.

[3] Y. Huang, C.R. Garrett, I. Ting, S. Parascho, and C.T. Mueller. Robotic additive construction of bar structures: unified sequence and motion planning. *Contr. Rob.*, 5:115–130, 2021.

[4] Stefana Parascho, Isla Xi Han, Samantha Walker, Alessandro Beghini, Edvard P. G. Bruun, and Sigrid Adriaenssens. Robotic vault: a cooperative robotic assembly method for brick vault construction. *Contr. Rob.*, 4:117–126, 2020.

[5] L. Nägele, A. Hoffmann, A. Schierl, and W. Reif. LegoBot: Automated planning for coordinated multi-robot assembly of LEGO structures. In *IEEE/RSJ Int.*

---

[4] The link to the simulation video:Video URL

Conf. Int. Robots and Syst. (IROS), pages 9088–9095, 2020.

[6] S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Pearson Education Limited, 2016.

[7] J. Relefors, M. Momeni, L. Pettersson, A.V. Papadopoulos, and T. Nolte. Installation order in automatic fabrication of reinforcement rebar cages. In *IEEE Conf. Emerging Tech. and Factory Aut. (ETFA)*, pages 1–4, 2021.