Automated Fabrication of Reinforcement Cages Using a Robotized Production Cell

Mahdi Momeni[a,*], Johan Relefors[a,b], Abubakar Khatry[c], Lars Pettersson[c],
Alessandro V. Papadopoulos[a], Thomas Nolte[a]

[a] *Mälardalen University, Västerås, Sweden*
[b] *Robotdalen AB, Västerås, Sweden*
[c] *Skanska Sweden AB, Stockholm, Sweden*

Abstract

Unlike what is common in the traditional manufacturing industry, the structures in the construction industry are often one of a kind. The goal of this paper is to provide a real-world compatible fully automated gantry-robot system for flexible serial production of custom-made reinforcement cages. This can lead to increased efficiency, productivity, and sustainability, not to mention the positive impact on labour safety as well as decreased environmental impact. In this paper, we present a solution utilizing three industrial robots mounted on a gantry structure, and for the automatic generation of robot paths for moving, placing, and tying rebars. Moreover, we present how a CAD model of a rebar cage, created in Tekla, along with installation instructions such as installation order, are transferred into CoppeliaSim. This proof-of-concept implementation is an important milestone indicating the feasibility of our proposed robotic solution for the automated construction of one-of-a-kind reinforcement cages.

Keywords: Automatic Construction, Digital Fabrication, Construction Robot, Path Planning, Reinforcement Cages

---

1. Introduction

In the context of the construction industry, installation of reinforcement bars (rebars) for concrete structures is a time-consuming manual process, performed one rebar at a time, see Figure 1. Traditionally, reinforcement is installed directly in the formwork, bar by bar. Prefabricated reinforcement cages are also

---

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

[*] Corresponding author

Email addresses: `mahdi.momeni@mdh.se` (Mahdi Momeni), `johan.relefors@robotdalen.se` (Johan Relefors), `abubakar.khatry@skanska.se` (Abubakar Khatry), `lars.g.pettersson@skanska.se` (Lars Pettersson), `alessandro.papadopoulos@mdh.se` (Alessandro V. Papadopoulos), `thomas.nolte@mdh.se` (Thomas Nolte)

Preprint submitted to Automation in Construction *October 11, 2021*

sometimes used. In such cases, the cages are often prefabricated off-site and transported to the site where they are installed in their permanent position. Either way, the installation of the bars is done manually.

This manual work poses several challenges to the workers that perform the work as well as to the projects that include this type of work. Two important challenges that can be mentioned are that reinforcement installation is very heavy and it is also time-consuming. We normally find reinforcement installation on the critical path of the projects. This type of work affects the worker's health and it also negatively affects project schedules. On top of this, the installation of reinforcement is normally found on the critical line of virtually all large civil construction projects, meaning that the project schedule is directly affected by the installation time. One way of dealing with the time-critical aspect of reinforcement installation is to manually prefabricate the reinforcement in bigger units, often referred to as reinforcement cages or *rebar cages*. These units can be fabricated at locations other than their final install locations and then, once fabricated, lifted into their permanent positions before pouring the concrete. This way of installing shortens the overall construction time, but will not change the fact that the reinforcement is still manually fabricated.

Figure 1: Manual reinforcement installation in progress.

The solution proposed in this paper has the benefit of minimizing the produced waste, as, in the proposed set-up, the reinforcement bars are bent and cut using bar coils. As a result, only the necessary reinforcement for the particular cage under construction is used, in contrast to the manual installation process, where bars not used for structural purposes are normally cut from standard length bars, leading to unnecessary waste on the site. Moreover, as the installation will be based on a 3D BIM model, only the necessary bars included in the design will be installed, avoiding further waste.

Moreover, the two most relevant advantages are the time saved on the critical path of the project, thanks to overall increased efficiency and productivity, and the reduction of hard work and wear of the workers required to install the reinforcement manually. In addition to that, removing the difficulties of manual fabrication is one of the benefits of such a process. Another benefit would be potential cost savings due to a faster fabrication process. However structures, like houses, bridges, or tunnels are unique and one of a kind, i.e., they are built for a special purpose, taking into account the site conditions when designing them. Even though repetitive operations are strived for, automating the reinforcement process for built-specific structures poses a different challenge from automating serial production. Most strikingly, each unique reinforcement cage requires its own set of operations to be performed. This means that the automation must be flexible enough to handle a large set of operations, enabling the fabrication of a large set of rebar cages.

In order to determine which operations are to be performed, all the information regarding how the rebar cage should be fabricated must be fed to the automation system, preferably as a digital model. One example of such a digital model is a three-dimensional CAD including detailed information for all rebars, see an example in Figure 2.

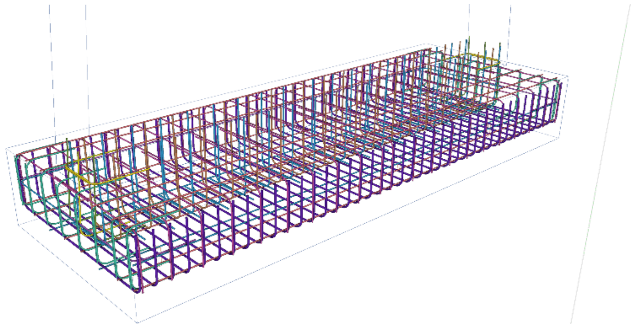


Figure 2: Digital drawing of a rebar cage.

In this paper, we investigate the automation of the fabrication of rebar cages. While we mention aspects of digitization, our main focus is on executing operations that, when executed in sequence, enable the fabrication of a specific rebar cage. To do this we present a production cell with a gantry structure and three industrial articulated robots. We also present algorithms for generating paths for the production cell which execute the required operations.

To demonstrate the capabilities of the presented solutions, we utilize a virtual production cell implemented in CoppeliaSim [1], with a use case for the assembly of a rebar cage provided by the construction company Skanska[1]. The demonstration covers key steps necessary, from drawings of the rebar cage to tying of individual rebars positioned in place by the industrial robots of the production cell. Moreover, we also outline some of the challenges that should be tackled for bringing the idea of robotic fabrication to real life. A simple sketch of

---

[1] `https://www.skanska.se/en-us/`

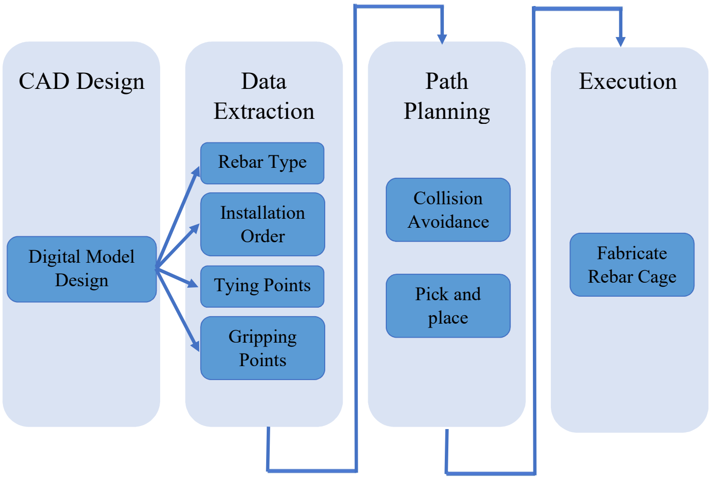


Figure 3: Overall process scheme.

the process for automatic fabrication of reinforcement cages is depicted in Figure 3. However, note that this sketch only intends to show the steps needed for automated pre-fabrication of the rebar cages, given the CAD model as an input and the corresponding pre-fabricated rebar cage as an output. This should be compared with other methods of pre-fabrication. The pre-fabricated cage will, just as any pre-fabricated cage, have to be transported to the site and installed in its permanent location. The transportation, as well as other changes in the construction process, which are needed to introduce such a robotic system, are well beyond the scope of this paper. While we mention aspects of digitalization and automation, our objective is focused on executing operations that, when executed in sequence, enable the fabrication of a specific rebar cage. It should also be noted that in the suggested solution, the robots are not supposed to move around the construction site but rather, their movements are limited to the place where the rebar cages are to be pre-fabricated.

The outline of the paper is as follows. Section 2 presents related work. Section 3 describes the production cell used for the automated fabrication of rebar cages. Section 4 presents an overview of the suggested approach for automatic fabrication of rebar cages. Section 5 outlines the path planning methods used for picking, placing, and tying rebars, followed by a proof-of-concept implementation in Section 6 and demonstration in Section 7. Finally, Section 9 concludes the paper and presents the discussion along with the future direction of this research.

2. Related Work

In comparison to advances in methods and technology used in the manufacturing industry, the methods and tools used in the construction industry have not developed significantly. Traditional methods used in today's construction industry don't necessarily address the demands for efficiency, productivity, and sustainability. Robotics is a key and enabling technology for innovation in the construction industry. Despite all the benefits offered by robotics, possible future scenarios of the application of robotics in the construction industry have not been systematically and extensively explored. Thus, how the construction industry will develop in the future and what kind of robots should be developed is not yet clear. Furthermore, because construction robotics is not yet an established field, design experiences or robust data from previous developments and applications are lacking [2].

In [3], one of the first research ideas on robotic construction is presented. In the paper, the possibility of using industrial robots and possible applications in the construction industry was investigated. In [4], the author presents the idea of using a gantry-robot system as well as the design of a modular robot system for building applications. In [5], the authors propose a robotic cell to assemble rebar cages for beams and columns. Their proposed robotic cell includes four industrial robots hanging on a gantry system. The inputs to this system are the pre-manufactured rebars, and the output consists of rebar cages, similar to what we present in this work. However, their paper only presents the proposed robotic cell without really building and testing it. In our case, we built the gantry-robot system, simulated and experimentally demonstrated the fabrication. We have also observed challenges when working to bring this idea to life, these will be discussed in the following sections. Additionally, we have also previously presented our proof-of-concept downscaled preliminary production cell for automated construction of rebar cages[2] and in [6] we have highlighted key challenges when it comes to automating the overall process of using the production cell to fabricate rebar cages.

In recent years, the scientific community has witnessed an increasing interest in 3D printed concrete [7] and robotic fabrication with concrete [8]. However, automated integration of structural reinforcement is still a challenging problem [9]. Hack et al. [10] present a robotic fabrication process that unifies concrete formwork and structural reinforcement. Kontovourkis and Tryfonos [11] developed a parametric-integrated algorithm for tool-path planning and 3D printing control using an industrial robot.

To the best of our knowledge, there does not exist any robotics system for the automatic fabrication of reinforcement rebar cages. However, there are two robots called TyBot as well as IronBot[3]. Yet, these robots are of a different nature and with different tasks. TyBot is a mobile tying machine/robot which ties the already placed rebars together, while the IronBot is a mobile pick and place single robot. These robots are meant to build cages in, e.g., bridge construction. None of these robots is able to cooperatively and automatically place the rebars one after another and tie them together to build cages. However,

---

[2] `https://www.youtube.com/watch?v=o9TOAa5g_Zo`

[3] https://www.constructionrobots.com/

in our case, the robots are interacting with each other to cooperatively mount different bar types one after another to fabricate a rebar cage corresponding to given CAD model as input.

From a robotics perspective, the problem that we face in this paper is to plan for single-piece manufacturing using three kinematically redundant gantry-robot structures. More specifically, the problem is divided into (i) planning in joint space for tying rebars together, and (ii) planning for bi-manual manipulation in Euclidean space for placing rebars in the rebar cage, which is a planning problem for a closed kinematic chain.

In [12] a similar problem to our placement planning problem is described. The authors describe the manipulation of parts made of carbon fibre reinforced plastics, to be used as aeroplane components. They use two 6 Degrees Of Freedom (DOF) robots, each on a linear axis, to manipulate a large number of different parts. Their path planning technique uses one robot as a master, for which a path is computed first, and the other robot as a slave which adapts to the path of the master. Their objective is to minimize programming time due to a large number of parts. In this work, however, we focus on the minimization of programming time for single-piece production.

Another work where bi-manual manipulation and a single axis are used in [13]. Their application is moving aluminium parts for an avionics application. It is worth noticing that they define the gripper positions manually. Similarly, we also define the gripping points as well as the tying points manually during the data extraction from the digital model. However, part of our future work will be to further investigate how to automatically generate the gripping and tying points.

We have not used the methods cited above for our bi-manual manipulation path planning. The main reason is that we were not aware of these methods when the work started. As a general comment on planning for a single robot at a time, this is generally not a good idea when the robots are likely to interfere with one another which is often the case in our path planning problem.

It is worth noticing that our planning problem must be solved under large uncertainties due to imperfections coming from the rebar manufacturing process, as well as deflections of rebars and parts of the gantry-robot system. Dealing with such non-idealities is an important problem, but it is beyond the scope of this paper, and it is left to future work. In this paper, we assume that everything is ideal, as explained in Section 4.

3. The Production Cell

We consider a production cell consisting of a gantry structure, with three robotic arms. A previous version of the production cell is shown in Figure 4. The three robotic arms are attached to the gantry structure from their base, and hanging downwards, i.e., they are positioned upside down. Labelling the vertical axis as the z-axis the gantries can move the base of each robot in the x and y directions. The setup uses three ABB IRB1200-7/0.7 industrial robots, although any 6 DOF robotic arm with similar kinematics will work. Combining the DOF of a single gantry-robot system gives a total of 8 DOF. The full system, consisting of three gantry-robot systems, has 24 DOF which, disregarding collisions, can be controlled independently.

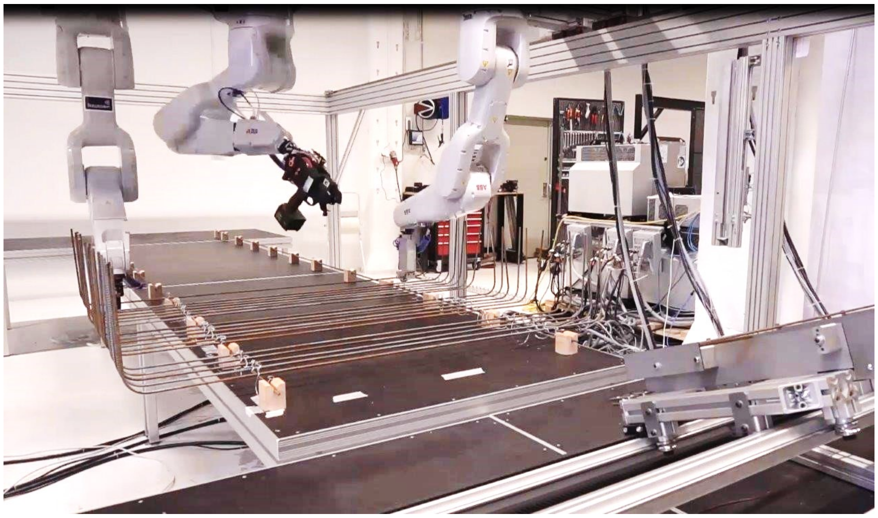


Figure 4: The down-scaled system with 2 DOF per gantry.

In our future production cell, and in the cell which we have used in simulation as well as the accompanying video for this paper, the gantry will be able to move the base of the robot also in z direction, giving 3 additional DOF to the gantry. The total number of DOF for the full gantry-robot system will then increase to 27. The methods proposed in this paper are developed for such a system. In the following, we refer to the full machine as the *full system*, or *system*, while each of the gantry-robot systems is referred to as a gantry-robot system. Note that the system intended for use in large civil construction projects, the full-scale system, will be much larger than the systems discussed above.

The production cell needs a delivery system for rebars. The full-scale system consists of a cut and bend machine and a fixture. The cut and bend machine takes rebar rolls and produces different rebar types as in Figure 5, according to a given order. The produced rebar is then placed in a fixture where the robots can pick them up. We have yet to develop this part of the real physical production cell. Instead, since we do not suffer the logistical problems of a real-world production cell in our simulation, we present the rebars as hanging in free space above a table. In the following, a rebar being in the fixture refers to the rebar being in the pick location.

4. Overall Approach

In this section, we briefly discuss our overall approach to the problem of going from a CAD model of a rebar cage to a finished rebar cage in the real world. We then define the scope of this paper as part of this overall approach and give an overview of related challenges. Specifically, we focus on and highlight details of picking, placing, and tying of rebars, along with the digitization process needed to facilitate such an automated installation of rebar cages.

4.1. Working Assumptions

The presented approach is based on the following working assumptions:

Assumption 1. *The gantry is a rigid structure, i.e., it does not deflect or twist because of the weight of the robots and the lifted rebars.*

Assumption 2. *The robot links and joints are not affected by physical loads, i.e., the kinematics of the robots are always described by the same equations.*

Assumption 3. *The rebars are ideal, i.e., they are rigid bodies, there are no deviations, deformations, or deflections*[4]*. In other words, the rebars are assumed to have their theoretical geometry.*

Assumption 4. *The tying tool produces tight enough knots to firmly tie the rebars together. In other words, after being tied together the rebars are fixed with respect to each other.*

Assumptions 1–4 are at the basis of the construction design, and they will be used in the simulation. Future research will be dedicated to investigating how to relax some or all of these assumptions, to port the proposed solution to a real-world solution.

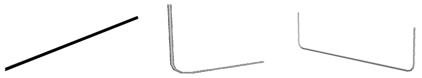
4.2. CAD to real-world cage

Given any method of assembly, there are constraints on the rebar cages that can be assembled. This in turn means that the design of rebar cages must be adapted to the method of assembly, whether automated or manual. Assuming that we have a rebar cage that is suitable for assembly in our production cell, the problem of going from a cage's digital model to the real-world assembled cage can be divided into the following steps:

1. Calculate an installation order based on which fabricating the cage is feasible

---

[4] By deviation, we mean their length is exactly as the nominal value. By deflection we mean they do not bend because of the gravity while lifted and moved by the robots. And by deformation or twist, we mean their shape doesn't change before/after fed to the cut and bend machine and/or during transportation to the site.



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

Figure 5: The rebar types that we consider in this paper.

2. Calculate grip points and tie points for the robots.
3. Generate system trajectories, which are compatible with the production cell.
4. Upload the generated instructions to the production cell's controller and execute.

Our overall goal is to start with a valid (from an automation perspective) digital model of a rebar cage as input and end up with an assembled rebar cage. In this paper, however, we mainly focus on the third step. More specifically, we take as input a CAD model of a rebar cage along with a file (supplementary data) containing the following information.

- **Installation order.** The order in which the rebars should be installed in the cage.
- **Rebar type.** The type of each rebar in the rebar cage. Different types of rebars exist, e.g., "A-bar", "B-bar", and "C-bar", depicted in Figure 5.
- **Rebar dimensions.** Geometrical properties of the rebars.
- **Rebar's placement location.** The target position for each rebar in the cage under construction.
- **Grip points.** The grip position for each of the gripping robots for each of the rebars.
- **Tie points** The positions where each rebar needs to be tied into the already built part of the cage.
- **Tie instructions.** Instructions for tying each rebar in the cage. Includes the order of the tie points, as well as information on when one or more of the placing robots must release the rebar for a tie point to be accessible.

The availability of this information is essential for the automation of the fabrication process. Future research will focus on the automated generation of most of the required supplementary data.

Algorithm 1 Procedure for the construction of the structure from the CAD model.

```
procedure BuildCage(Robots,CAD_model)
    PlacePaths, TiePaths, InstallationOrder ← ComputePaths(Robots, CAD_model)
    Cage ← ∅                                        ▷ Initialize the cage to have no rebar
    while InstallationOrder ≠ ∅ do                  ▷ Until there are no rebars to place
        rebar ← InstallationOrder.next()            ▷ Get rebar to place
        path ← PlacePaths.next()                    ▷ Get the path to place next rebar
        actuate_path(Robots,path)                   ▷ Put the rebar in place
        Cage.add(rebar)                             ▷ Add the rebar to the digital twin of the cage
        path ← TiePaths.next()                      ▷ Get the tie paths to place next rebar
        actuate_path(Robots,path)                   ▷ Tie rebar into cage
        goto Homing position
    end while
end procedure
```

4.3. Automated fabrication of the rebar cage

We consider a production cell as described in Section 3. The objective is to start from a 3D representation of the rebar cage to be constructed, and automatically generate a plan for the full system that, when executed in the production cell, manufactures the rebar cage. An overview of the proposed method for accomplishing this is presented in Algorithm 1.

A model of the production cell and the CAD model are the inputs to Algorithm 1. These inputs can be used for a digital twin, which will also allow us to keep track of the current status of the production cell and assembly during operation. Information such as robot grips, where to tie, installation order, and robot paths, is to be generated. At this point, we use supplementary data for grips, tie points, and installation order and generate only the robot paths. In our case, the Tekla[5] software was used to generate the CAD model.

Note that the required data to control the robot cell are related to:

1. The identification of tie points and potential gripping points, based on the type of rebars.
2. The computation of the installation order, which grip points to use, and ordering of tie points.

Such a problem is more complex, and the adoption of Artificial Intelligence (AI)-based solutions for automated planning could be beneficial to explore [14, 15, 16]. However, AI-based methods will be explored in the future as a viable solution to further speed up and possibly optimize the overall process.

The first step in the algorithm is to calculate the installation order and the gantry-robot paths for installing all the rebars. The computation of the gantry-robot paths is as described in Algorithm 2, while the installation order is taken from supplementary data. We then follow the installation order and actuate

---

[5] `https://www.tekla.com/`

Algorithm 2 Algorithm for computing the paths needed to build the structure from the CAD model.

```
function COMPUTEPATHS(Robots,CAD_model)
    PlacePaths ← ∅                                          ▷ Initialize the paths for placing rebar
    TiePaths ← ∅                                             ▷ Initialize the paths for tying rebars
    Installation_order ← compute_installation_order(Robots, CAD_model)
    Cage ← CAD_model                          ▷ Initialize the cage to have the full CAD_model
    while Cage $\neq \emptyset$ do                                  ▷ Until the cage is not de-constructed
        rebar ← Installation_order.last()                        ▷ Get the last rebar to be placed
        hold_pos ← find_hold(CAD_model,rebar)    ▷ Get the hold positions of the rebar
        path ← compute_path(Robots,Cage,rebar,hold_pos)
        PlacePaths.insert_at_beginning(inverse(path))       ▷ Store the inverse of the path
        tie_pos ← find_ties(CAD_model,rebar)
        while tie_pos $\neq \emptyset$ do                              ▷ Get the tie positions for the rebar
            path ← compute_tie_path(Robots,Cage,rebar,tie_pos)
            TiePaths.insert_at_beginning(path)                              ▷ Store the tie path
            Cage.remove(rebar)              ▷ Remove the rebar to the digital twin of the cage
        end while
        goto Homing position
    end while
return PlacePaths, TiePaths, InstallationOrder
end function
```

the gantry-robot systems to follow each of the paths for placing and tying the rebars. Once the rebar has been placed, the rebar is tied into the structure.

Finally, the gantry-robot systems are moved to their respective homing position, waiting for new rebar to be produced which is then placed and tied in the same way as the previous rebar.

4.4. Computation of the robot paths

Computing the paths while assembling a cage is time-consuming and impractical. In our approach, we work in a digital model of the production cell and we “de-construct” the cage. This means that we compute paths for untying and picking rebars from a cage and placing them in the fixture. To manufacture the cage in the real production cell, untying becomes tying and the paths, as well as their order, are reversed.

The rationale for working backwards is that once the rebar is far enough – but not too far – away from the cage, the number of valid robot configurations is much larger than when the rebar is at its final location. This indicates that if we start from valid configurations for placing and tying the rebars, the problem of finding a path to the fixture should be easier than if we started from valid configurations at the fixture and tried to find a path for placing and tying the rebar.

We use the provided installation order to determine which rebar to install. The first step is to determine the rebar path, from its location in the rebar cage to its place in a fixture. The next step is to determine possible “place and tie” configurations. These are picked randomly by sampling from the allowed configurations. Starting from the place configuration, we try to generate paths for the system to follow the generated rebar movement. Once that is done, tie paths are generated by again setting the rebar in the cage and then planning tie paths, possibly letting go of the rebar in order to access all tie points. If either the generation of the place path or the tie paths fails, we go back and try to generate new place and tie configurations.

When paths have been generated for picking, placing, and tying a rebar, that rebar is removed from the cage. The next rebar in the installation order is then addressed, with a slight modification. We try to reuse any previously generated path. This is done by offsetting the paths for the active robots[6] to match the new rebars position and checking that the resulting pick, place and tie operations can be performed with no collisions.

5. Path Planning

In this section we first outline the path planning method that we use for picking, placing, and tying the rebars and then we give a more detailed description of the steps involved.

The path planning is conducted through the execution of the following steps, each of which will be described in more detail later:

1. Create the path that the rebar will travel.
2. Create robot configurations for placing the rebar in the rebar cage.
3. Create robot configurations for tying the tie points while one or more robots are still holding the rebar.
4. Create the robot paths for pick and place of rebar.
5. Create paths for the robot tying rebars together while potentially moving non-tying robots out of the way.

The first step, where the rebar path is computed, is considered independently from the robot movement. This means that we assume that we can compute robot movements to execute the pick and place based on the computed rebar path. While this makes path planning easier, it also limits which rebar cages can be built using our method[7]. Finally, note that the order of the last two steps can be switched in the method that we use.

5.1. Creating the rebar path

The rebar path is computed starting from the rebar's placement position in the cage and ending with the rebar in the fixture. The reason for this reverse

---

[6] At times one or more robots are not involved and these must then simply be kept out of the way during the motions. We do this by defining safe locations for each of the robots, where the safe location for the middle robot depends on the current location of the two outer robots.

[7] Relaxing this assumption is part of our future work.

operation is that placing the rebar in the fixture is an easier problem than placing the rebar in the cage, mainly due to the crowded environment in the rebar cage.

To create a path that removes a rebar, called the moving rebar below, from the rebar cage, we use a number of deterministic steps. The pose of the moving rebar is stored after each step. This gives a path consisting of waypoints. The steps are:

1. Apply an analogue of a repulsive potential between the moving rebar from its neighbours in the cage.
2. Lift the rebar straight up to a height where it can move freely above the cage.
3. Translate the rebar to the middle of the cage
4. Translate and rotate the rebar such that it is positioned above the rebar fixture.
5. Lower the rebar into the fixture.

Applying a repulsive potential between the moving rebar and the other rebars in the cage means the moving rebar will be pushed away from its neighbours in a direction where no collisions occur. The idea is that this step should give an approach to the final location. This is essentially just potential field path planning, see for example [17], which introduces the possibility of getting trapped in a local minimum or oscillations. In this case, we assume that the rebar cage and the installation order are such that this does not happen.

Regarding the form of the potential, there are many possibilities and we have not searched for criteria to distinguish an optimal one. We have chosen a potential guided by the initial and desired distances between the moving rebar and the other rebars in the cage, as well as the number of iterations needed to remove the rebar. The chosen potential gives the offset magnitude

$$d(\Delta) = 0.002 \cdot \Delta \cdot \left(\frac{1}{\Delta} - \frac{1}{0.1} \right), \text{ when } \Delta \leq 10mm, \tag{1}$$

$$d(\Delta) = 0, \text{ when } \Delta > 10mm, \tag{2}$$

where Δ is the minimum distance between the two rebars. The potential is plotted in Figure 6. The direction of the contribution is given by the direction of the vector from one rebar to the other.

The displacement contributions from all rebars with respect to the moving rebar are added to get a total displacement. We then impose a minimum and maximum allowed displacement for each iteration. If the magnitude of the displacement is larger (smaller) than the maximum (minimum) the magnitude is set to the maximum (minimum). The chosen maximum is 2.5 mm and the minimum is 1 mm. We stop iterating once the minimum distance from the moving rebar to all other rebars is 2 mm.

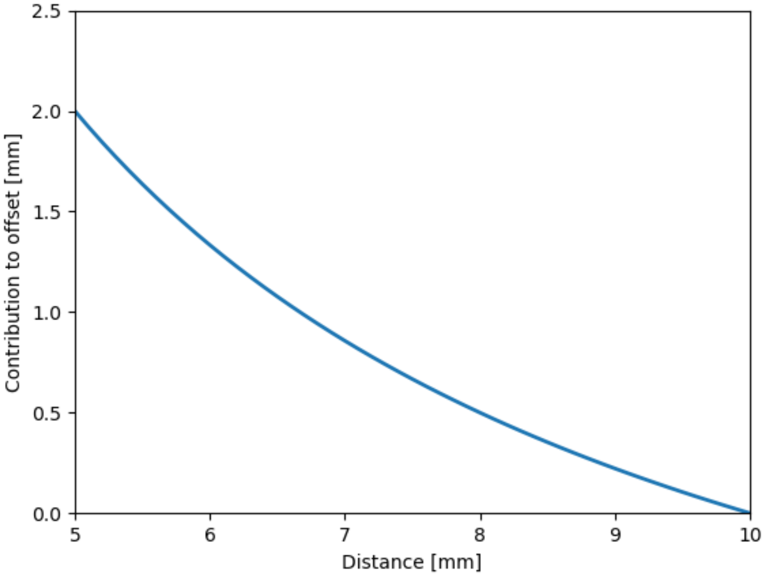


Figure 6: The offset contribution from a single rebar to a moving rebar. The distance refers to the minimum distance between the rebars. Note that the rebars start at least 5 mm apart which is why the x-axis starts at 5 mm.

Following the repulsion step, we lift the rebar straight up, while checking for collisions, to a height where the rebar is free to move. The rebar is then moved above the fixture with the same orientation as it needs to have in the fixture. Finally, the rebar is lowered to its position in the fixture.

5.2. Creating place configurations

The way that we create configurations for placing the rebar is to sample the gantries in different positions and perform Inverse Kinematics (IK) for the remaining 6 DOF. We then check that the robot can release the rebar in its final location without hitting any kinematic limits or obstacles. Any robot that is not used is placed in a location where it will not interfere with the place movement.

If tying the rebars was not an issue it would most likely be good enough to perform this process until one collision-free place configuration is found. We do however want to keep the robots that are placing the rebar still when tying, at least for a few of the tie points. Therefore we have to adapt the place configuration to give enough room for tying. To do this we sample several place configurations and select the one with the largest difference along the first gantry axis, the sampling is described in more detail in Section 6. This does not always ensure that good tie configurations can be found but it is a useful heuristic. Other possible heuristics might guide the choice of place configuration. One possibility is to look at the joint angles for the last 6 DOF and see that the joint limits are as far away from their respective limits as possible.

5.3. Creating tie configurations

As mentioned in the previous subsection, we keep the robots that are placing the rebar still when tying at least a few of the tie points associated with a particular rebar. Some tie points might not be reachable in this way, and we assume that when this is the case it is indicated by the input data to the path planner.

As long as we keep the placing robots still we simply sample the tying robot at the tie location, as we sampled for the place configurations. We also make sure that the tying robot can perform a linear approach movement. This process typically has to be repeated a few times in order to find valid tie configurations.

When we need to move one of the placing robots out of the way, it is placed in a suitable location where it will not interfere with tying. The sampling of the tying robot then continues as before.

5.4. Pick-and-place path

What goes into this step is the rebar waypoints as well as the place configuration, which is the configuration of the system in the first waypoint. We then sample the full system at the next rebar waypoint, as before by sampling the gantry and using IK for the last 6 DOFs (the IK step is only to check for reachability as will be explained below). To get the path in between the waypoints we interpolate the rebar and the gantry-robot system together.

The rebar is interpolated using spherical linear interpolation (SLERP) [18] for orientation and linear interpolation for the position. For the robot interpolation, the gantry is interpolated linearly while the last 6 DOFs are determined using IK in order to follow the grips on the rebar and that the path must be continuous in the robot joints.

Once a path between the first and second rebar waypoint is determined the process continues between the second and third rebar waypoint, using the computed configuration at waypoint two as the starting point for the robots. The process typically does not work for every possible random sample at each rebar waypoint. For this reason, we try multiple times in order to find a path. When doing this we apply a depth-first search where each starting configuration at a given rebar waypoint is used for some maximum number of times and we stop searching once one path to the fixture is found.

Note that the algorithm presented above is not probabilistically complete. Meaning that it will not always find a solution, even in cases where there is one. However, we have not found a complete algorithm for this planning scenario.

5.5. Creating tie paths

In short, the tie paths are computed using bi-directional RRT-Connect [19, 17], referred to as BiRRT in the following. The tying robot starts in a configuration where it is out of the way for the placement of the rebar. We then plan

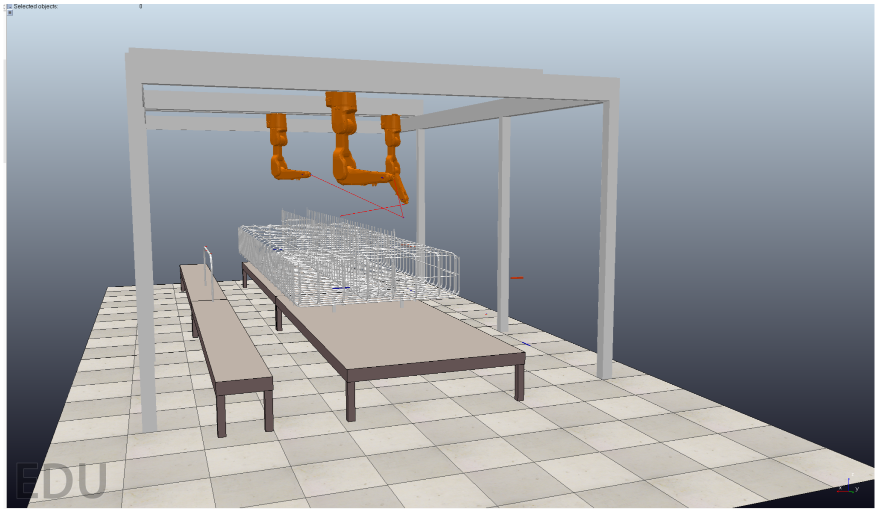


Figure 7: CoppeliaSim simulation scene used for virtual demonstration.

to the first tie point, then the second, and so on. If any of the place robots need to be removed for tying to continue then the path for this movement is also planned using a BiRRT while keeping the other two robots standing still.

6. Implementation in CoppeliaSim

In order to test the algorithms described in Section 4 and Section 5 we have implemented them as a plugin to CoppeliaSim using its regular API[8]. In CoppeliaSim we have set up the gantry structures of the production cell of Section 3, as well as the robots. The gantry is modelled using cuboids for the links and we have then added joints to be able to move the industrial robots around using the gantry. The robots that we use have been implemented by importing ABB IRB1200-7/0.7 CAD data[9] to CoppeliaSim and adding the proper joints to the model. The setup is displayed in Figure 7.

In the following, we describe implementation details concerning the central steps of (i) data import, (ii) sampling, (iii) interpolation, and (iv) path planning, respectively.

6.1. Data import

This subsection details the central steps involved when loading all relevant data of the rebar cage to be constructed by the production cell into the memory of CoppeliaSim. In addition, as part of the data being automatically imported into CoppeliaSim, in this paper the installation order (of rebars) is given in the

---

[8] see https://www.coppeliarobotics.com/

[9] found from `https://abb.com/`

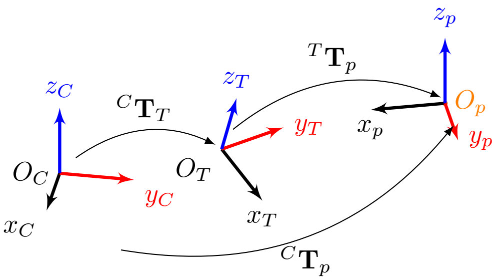


Figure 8: To transfer frames given in the Tekla reference frame, O_T, to the CoppeliaSim reference frame, O_C, we need to identify the transform between the frames, $^C\mathbf{T}_T$. O_p is a frame that is fixed in the CAD model of the rebar cage. By expressing the transformation to O_p in Tekla, giving $^T\mathbf{T}_p$, and in CoppeliaSim, giving $^C\mathbf{T}_p$, the frame desired transformation $^C\mathbf{T}_T$ can be determined.

data file. The installation order is the sequence order for which the assembly process of the cage's rebars is supposed to be executed, to eventually end up with the desired reinforcement cage. The installation order must be, and it is assumed to be, a feasible order with respect to assembly in the production cell.

6.1.1. Coordinate transformation

Rigid objects in space can be described by their position and orientation. The combination of position and orientation is usually referred to as the *pose* of the object. A *frame* is a Cartesian coordinate system, i.e., a set of orthogonal axes which intersect at a point known as the *origin*. The pose of an object is expressed in a reference frame that uniquely identifies the object in space.

All frames in space in the extracted data from the Tekla digital model, including the gripping and tying frames are with respect to the reference frame in the Tekla model. Hence, for these frames to refer to the corresponding frames in the CoppeliaSim model, it is necessary to develop a transformation matrix [20] between the Tekla reference frame and the CoppeliaSim reference frame.

The homogeneous transformation matrix, $^C\mathbf{T}_T$, is a 4×4 matrix that maps frames defined with respect to the Tekla reference frame to the corresponding frame with respect to the CoppeliaSim model reference frame, see Figure 8. The transformation matrix is defined as follows

$$^C\mathbf{T}_p = {^C\mathbf{T}_T} \cdot {^T\mathbf{T}_p} \quad \Rightarrow \quad {^C\mathbf{T}_T} = {^C\mathbf{T}_p} \cdot {^T\mathbf{T}_p}^{-1} \tag{3}$$

where $^C\mathbf{T}_p$ and $^T\mathbf{T}_p$ are the 4×4 matrices representing a frame in the CoppeliaSim coordinate frame and the Tekla coordinate frame, respectively.

6.1.2. Calculation of $^C\mathbf{T}_T$

Looking at Eq. (3), we can find $^C\mathbf{T}_T$ by identifying $^C\mathbf{T}_P$ and $^T\mathbf{T}_P$ for some P which is fixed in the CAD model. In this paper, we have used an A-bar in the bottom of the cage since it gives a clear direction along the cage and we already know that the z-axes align.

6.1.3. Mapping our input

In our work, the data extracted from Tekla is structured in such a way that the orientation of each point is presented in rotations around the fixed x-axis, y-axis, and z-axis, i.e., α, β, and γ format. That is to say, for a more general case where the coordinate frames assigned to the desired frames are not aligned with the reference coordinate frame in Tekla, the $\mathbf{R}_{3\times 3}$ element of ${}^{C}\mathbf{T}_{p}$ is formulated as

$$\mathbf{R}_{3\times 3} = \mathbf{R}_{x}(\alpha) \cdot \mathbf{R}_{y}(\beta) \cdot \mathbf{R}_{z}(\gamma) \tag{4}$$

where $\mathbf{R}_{x}(\alpha)$, $\mathbf{R}_{y}(\beta)$, and $\mathbf{R}_{z}(\gamma)$ are the rotation matrices around x-axis, y-axis, and z-axis in the Tekla coordinate system, respectively. Then, any point with arbitrary position and rotation in 3D space in the Tekla digital model can be mapped to the corresponding point in the CoppeliaSim digital model by

$${}^{C}\mathbf{T}_{p} = {}^{C}\mathbf{T}_{T} \cdot \begin{bmatrix} \mathbf{R}_{3\times 3} & {}^{T}\mathbf{p} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{5}$$

where ${}^{C}\mathbf{T}_{T}$ is the pose of the desired point with respect to the Tekla and CoppeliaSim reference frames and ${}^{T}\mathbf{p} = [{}^{T}x_{p}\, {}^{T}y_{p}\, {}^{T}z_{p}]^{\top}$ are the x, y, and z coordinates of the point with respect to the Tekla reference frame, respectively as in Figure 8.

6.2. Sampling

We have implemented two types of random sampling. The types are i) sampling in joint space, which simply returns a position in configuration space, and ii) sampling in the workspace, where the gantry is sampled somewhere around the desired work-space position and the robot configuration is randomly selected among the available inverse kinematics solutions (if there are any).

When sampling the gantry, the distribution around the desired Tool Center Point (TCP) is not a uniform sphere for the first and third robot. Instead, we skew the distribution in the sphere towards a desired direction along the y-coordinate as well as that we make sure that the base of the robot is placed above the target. To get the base position we start from the target pose $[x_t, y_t, z_t]^{\top}$ and add a vector $[x, y, z]^{\top}$. To determine the vector to add, we sample in coordinates $[r, \phi, \Delta z]^{\top}$ where the relationship to $[x_t, y_t, z_t]^{\top}$ is given by

$$x = r \cdot \cos(\phi) \cdot \sqrt{1-(\Delta z)^2} \tag{6}$$

$$y = r \cdot \sin(\phi) \cdot \sqrt{1-(\Delta z)^2} \tag{7}$$

$$z = r \cdot \Delta z + baseHeight, \tag{8}$$

where *baseHeight* is the height of the robot base, from mounting point to the first joint. The sampling is done according to

$$r = X \times reach, \qquad \text{where } X \sim \mathrm{Beta}(1.5, 2.5) \tag{9}$$

$$\phi = Y, \qquad \text{where } Y \sim \mathrm{tri}(\phi_0 - \pi, \phi_0 + \pi, \phi_0) \tag{10}$$

$$\Delta z = Z, \qquad \text{where } Z \sim \mathcal{U}(0, 1) \tag{11}$$

where *reach* is the reach of the robot arm and ϕ_0 is the desired direction in the x, y-plane. Furthermore, we have used the probability distributions[10] (i) beta distribution $\mathrm{Beta}(\alpha, \beta)$, (ii) triangular distribution $\mathrm{tri}(a, b, c)$ centered around c with limits a and b, and (iii) uniform distribution $\mathcal{U}(a, b)$ between a and b. Note that if there is no prefered angle ϕ_0 the triangular distribution can be replaced with $\mathcal{U}(-\pi, \pi)$. We then determine $[x, y, z]^\top$ and the resulting base position is transformed into a gantry position where IK is performed to get a sample.

When sampling configurations in joint- and work-space with one or more systems, the other systems must also get a configuration. This situation happens, for example, when the tying robot is sampled to find a path between tie frames or when the placing robots are sampled to find a path for placing a rebar. In these cases, the robots which are not active in the movement must still have configurations. To implement this we have two other options for sampling. The first option is to stay still in a given configuration, as what is done by the placing robots when the tying robot is sampled for finding a tie path. The second option is to make sure that a robot is out of the way, as is done by the tying robot when the placing robots are sampled for placing the rebar. Regardless of how sampling is performed, we make sure that the system as a whole remains in a valid state, respecting the order of the different gantries and making sure that there are no collisions.

6.3. Interpolation

For the work described in this paper, we need three different modes of gantry-robot interpolation depending on the situation. The three different modes of gantry-robot interpolation are:

1. Interpolation in configuration space, where each joint variable, for robot and gantry, is linearly interpolated. This mode is used for example when tying.
2. Linear interpolation in the workspace, where the robot TCP is to follow a linear trajectory, while possibly changing its orientation. In this mode, a starting configuration for gantry and robot is supplied, as well as a goal gantry configuration. The gantry joints are interpolated linearly, while the robot joints are determined by IK along the trajectory. In cases where the orientation is changing along the trajectory, the orientation is determined using Slerp [18]. This mode is used for example when approaching a rebar.
3. Interpolation, where a rebar is being moved in the workspace and a gantry-robot system is moved to keep its TCP fixed in the rebar frame. In this mode, as in the linear mode, a start configuration for gantry and robot as well as a goal gantry configuration is provided, along with a rebar path and a grip position in the rebar frame. As the rebar is being moved in

---

[10] Note that in this paper the selection of distributions is a result of experimentation rather than of a formal evaluation.

the workspace the gantry is linearly interpolated between its start and end position while the robot joints are determined using IK. This mode is used when transporting a rebar.

Note that each of the three interpolation modes can be used for one or more of the gantry-robot systems, and different gantry-robot systems can be interpolated in different modes at the same time.

6.4. Path planning

In this paper, we have implemented two different types of randomized planners for use in different situations. One is a BiRRT [19, 17], which we use for planning in configuration space. This is used when planning motions between different tie frames, as well as when moving robots out of the way. The other planner is the one that is used for creating robot paths to place a rebar. We refer to this as the place planner below.

The job of the place planner is to create gantry-robot paths for a given rebar path. The rebar path is from the rebar's pose in the cage to its pose in the fixture, and it is given as a list of waypoints. Apart from the rebar path the place planner also takes grip poses relative to the rebar as well as place configurations as input.

From this input, the place planner uses a depth-first search strategy to find robot configurations along the rebar path. To do this the planner samples one set of gantry configurations for the second waypoint (workspace planning is used to make sure that the robots can reach the grip locations at the waypoint). Then, linear interpolation following the rebar is used to determine if the gantry sample can be used to generate a path for the whole system between the rebar waypoints. Once a path is found to a waypoint, the algorithm then goes for the next waypoint in the same way until hopefully reaching the goal. Note that for a given starting waypoint we try a fixed number of times to see if we can reach the next waypoint.

7. Simulation results

To determine whether the concepts that we have explained in the previous sections are feasible in CoppeliaSim or not, we have implemented them as shown in Figure 7. Our simulation results[11] verify that it is indeed possible to transfer the needed data from a Tekla digital model into a CoppeliaSim digital model. Moreover, we have successfully implemented the in this paper presented path planning algorithm for gripping and tying rebars. Hence, we were able to automatically fabricate the whole reinforcement rebar cage, presented in Figure 2, bar by bar, using the gantry-robot system under the given assumptions.

---

[11] A video-clip of the developed simulation scene is available at `http://www.idt.mdh.se/personal/aps01/research/robotics/constructionRobots.mp4`

8. Discussion

Figure 2 shows a CAD model of the reinforcement in a bridge support base slab. The bridge was built a few years ago close to Stockholm, Sweden. To investigate the feasibility of fabricating rebar cages using industrial robots the reinforcement in Figure 2 was built by robots in a 1:2 down-scaled laboratory version of the cage. Three ABB IRB 1200 robots were used for this purpose. Upon building the gantry-robot system and configuration of the gantry, the system has to be calibrated to minimize the positioning error. A method for this purpose was developed. The rebar cage digital model was imported into the ABB RobotStudio[12] software for simulation of the process as well as off-line programming of the entire gantry-robot system. In the laboratory, rebars of different types were placed on a fixture where two robots could cooperatively pick and mount them on their designated positions. The third robot, i.e., the tying robot, was then able to tie the rebars together while the two other robots were holding the rebar. This process continued and the rebars were mounted and tied together with one after another until the entire cage was built. During this demonstration, we noticed several challenges some of which are mentioned in this paper and are planned to be addressed in the continued and future work.

One major challenge was to find a way to calculate the robot movements. This was a necessary step since the structures are normally one of a kind. This part of the project is what is the main part of this paper. We decided to use the software CoppeliaSim[13] in this part of the work because we found that it is efficient in collision detection.

Another important challenge that we noticed during the demonstration was related to tying the bars together. The tool used for this purpose needs to generate tight and stable knots. Also, the interface between the robot and the tool needs to be arranged in such a way that the two can communicate in a good way. In co-operating with the Swedish company Husqvarna, we have come across such a tying tool. This tool will be used in the upcoming laboratory testing.

We have now modified the gantry-robot system enabling the robots to move, not just in the x-y plane but also in the vertical z-direction. This modification allows the system to build more complex rebar cages. The next step in the project will be to test this modified gantry-robot system together with the algorithms for automatic calculation of the robot movements described in this paper. This testing will again be in a laboratory environment and downscaled 1:2.

As explained in the previous sections, the presented robotic system is not a mobile platform in the sense that it is going to move around the construction site. This robotic system is complemented with a bar bending and cutting machine, and is going to be built somewhere on-site to pre-fabricate the rebar

---

[12] `https://new.abb.com/products/robotics/robotstudio`

[13] `https://www.coppeliarobotics.com/`

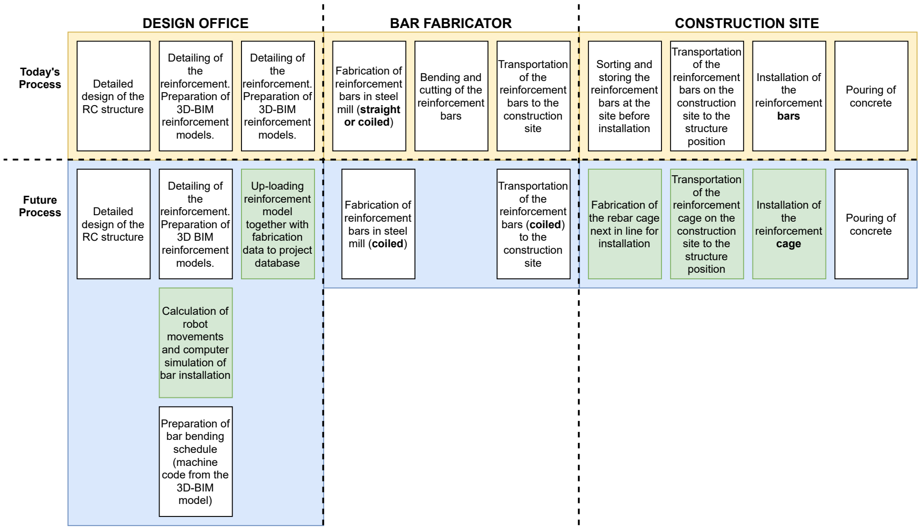


Figure 9: Comparison of the current process, and the future process with the automated solution.

cages. Furthermore, if for some reason it is not possible to build such a station on-site, the rebar cages can still be pre-fabricated off-site and transported to the site, as it is done in some construction projects already today, albeit that these cages are fabricated manually.

The deployment of robotic solutions on a construction site is complicated. Moreover, there are also issues regarding the impact on the construction process. The solution that we propose can be compared to pre-fabrication close to the construction site. This reduces the problem of having a robot or a system of robots moving around the construction site into the problem of moving reinforcement cages around the site, something which has been tried before. With this perspective, we chose to tackle the technical problem of rebar installation, to begin with.

The proposed approach can have a significant impact on the schedule of a construction project, and the end-to-end process. Figure 9 shows a summary of the different stages of a generic construction project, comparing the traditional approach (in the first row) to the envisioned future process. The current and future processes are based on [21, 22, 23], where the digitalization/automation of a construction process has been analyzed. The coloured cells of Figure 9 highlight the activities that will be either introduced or affected due to the proposed solution. Note that, in the future process, a more advanced level of digitalization/automation is required as the 3D-BIM model will be used to generate the robot program to fabricate the rebar cages.

9. Conclusion and Future Works

In this paper, we have presented how to transfer data (model transformation) from a Tekla model of a reinforcement cage (consisting of a set of rebars) to corresponding models in CoppeliaSim. Moreover, we have presented path planning algorithms for transporting and tying these rebars. Our proposed approach was under the assumption that the gantry, the robot, and the rebars are completely stiff and ideal. Our simulation results, under such assumptions, verify that the concept successfully operates as envisioned. Additionally, to get a better insight into the potential challenges in bringing to life such an idea, we decided to manually fabricate the exact cage used in the simulation. In this manual fabrication, we observed several challenges which are yet to be addressed by our algorithms in future research. We would call them "transition to the real world". These challenges include, but may not be limited to:

- automated extraction of the required data, e.g., gripping points and tying points, from the digital model of the rebar cage,
- automated generation of the rebar installation order, i.e., how to place the bars one after another such that building the cage is feasible,
- path planning for placing rebars and for tying rebars when they are not assumed to be ideal anymore, and
- adapting a tying tool to be used together with the industrial robots.

For the simulation described in this paper, we have extracted the data from the Tekla model manually. This is a very tedious task given the fact that the reinforcement cages normally include many bars, e.g., the model that we have used for the simulation described in this paper includes more than 200 rebars. In addition to extracting the data, for fully automatic fabrication of the reinforcement cage, it is necessary to automatically generate a valid order for placing the rebars one after the other such that fabrication of the reinforcement cage is indeed feasible, i.e., without any collision. Finding such an order is a challenging problem given the fact that there is a large number of rebars to place, meaning that a pure combinatorial search is not possible and/or extremely time-consuming. Hence a heuristic solution should be implemented.

The path planning algorithm described in this paper assumes that the bars, the gantry, and the robots are ideal and stiff. However, in the real world, the rebars, the gantry, and the robots are more or less flexible. Moreover, they likely deflect due to gravity. The deflection of the rebars, being relatively big, must be taken into account in the collision checking algorithm. In addition to that, the rebar deviations with respect to their theoretical geometry will affect their final positions in the cage which need to be considered when tying the bars together.

After placing a rebar in its correct position, it must be tied into place using a tying tool. The tying tool which we are currently using is a tying tool meant for manual tying which later has been adapted for robotic applications. Several important aspects should be taken into account with any tying tool: (i) quality and tightness of the knots, (ii) tolerance of the tool meaning that how close the two bars should be to each other for the tying tool to be able to tie them together tightly, and (iii) giving a reliable feedback signal such that we know that the bars are tied tight enough together and the process has finished properly. The tightness of the knots is, however, of particular importance. Based on our observations, if the rebars are not tied together tight enough, they will displace from their position. This displacement, in return, can cause some serious problems. These problems are, but not limited to:

- Tying failure because of the tying point displacement. In other words, it won't be possible to tie the next rebar to the already displaced rebar from the pre-defined tying point, and
- Collision of the robot and/or the next rebar(s) with the displaced rebar.

In our simulation, we have yet to equip the robots with tools, as these tools are still in development. The idea, however, is to use the middle robot for tying and the two other robots for pick and place of the rebar. We are also looking into developing tools that work for both gripping and tying which would make it possible to use all three robots for either task. Another way to increase flexibility in terms of what each robot can do is to use a tool changer for one or more robots.

In summary, our future work will include, but may not be limited to a) Automatic generation of the required data, b) Dealing with deviations of the robot, gantry, and the rebars from their theoretical geometry, c) Investigating the possibility of modifying the digital model and considering the addition of a stable frame so that the rebars are fixed in their final position, even when the knots are not tied enough.

References

[1] E. Rohmer, S. P. N. Singh, M. Freese, V-rep: A versatile and scalable robot simulation framework, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 1321–1326. `doi:10.1109/IROS.2013.6696520`.

[2] M. Pan, T. Linner, W. Pan, H. Cheng, T. Bock, Structuring the context for construction robot development through integrated scenario approach, Automation in Construction 114 (3) (2020) pp. 10–21. `doi:10.1016/j.autcon.2020.103174`.

[3] A. Warzsawski, D. A. Sangrey, Robotics in building construction, Journal of Construction Engineering and Management 111 (3) (1985) pp. 260–280. `doi:10.1061/(ASCE)0733-9364(1985)111:3(260)`.

[4] W. Leyh, Experiences with the construction of a building assembly robot, Automation in Construction 4 (1) (1995) pp. 45–60. `doi:10.1016/0926-5805(94)00034-K`.

[5] B. Dolinšek, J. Duhovnik, Robotic assembly of rebar cages for beams and columns, Automation in Construction 8 (2) (1998) pp. 195–207. `doi:https://doi.org/10.1016/S0926-5805(98)00083-1`.

[6] J. Relefors, M. Momeni Kelageri, L. Petterson, E. Hellström, A. Thunell, A. V. Papadopoulos, T. Nolte, Towards automated installation of reinforcement using industrial robots, in: IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 2019, pp. 1595–1598. `doi:10.1109/ETFA.2019.8869343`.

[7] A. Abou Yassin, F. Hamzeh, F. Al Sakka, Agent based modeling to optimize workflow of robotic steel and concrete 3D printers, Automation in Construction 110 (2020) 103040. `doi:10.1016/j.autcon.2019.103040`.

[8] P. F. Martins, J. Sousa, Digital fabrication technology in concrete architecture, in: Proceedings of the 32nd Education and Research in Computer Aided Architectural Design in Europe Conference (eCAADe), 2014, pp. 475–484, (Date accessed: 28/04/2021).
URL `http://papers.cumincad.org/cgi-bin/works/Show?ecaade2014_239`

[9] D. Asprone, C. Menna, F. P. Bos, T. A. Salet, J. Mata-Falcón, W. Kaufmann, Rethinking reinforcement for digital fabrication with concrete, Cement and Concrete Research 112 (3) (2018) pp. 111–121. `doi:10.1016/j.cemconres.2018.05.020`.

[10] N. Hack, K. Dörfler, A. N. Walzer, T. Wangler, J. Mata-Falcón, N. Kumar, J. Buchli, W. Kaufmann, R. J.Flatt, F. Gramazio, M. Kohler, Structural stay-in-place formwork for robotic in situ fabrication of non-standard concrete structures: A real scale architectural demonstrator, Automation in Construction 115 (3) (2020) pp. 103197. `doi:10.1016/j.autcon.2020.103197`.

[11] O. Kontovourkis, G. Tryfonos, Robotic 3D clay printing of prefabricated non-conventional wall components based on a parametric-integrated design, Automation in Construction 110 (3) (2020) pp. 103005. `doi:10.1016/j.autcon.2019.103005`.

[12] L. Larsen, M. Kaspar, A. Schuster, M. Vistein, J. Kim, M. Kupke, Full automatic path planning of cooperating robots in industrial applications, in: IEEE Conference on Automation Science and Engineering (CASE), 2017, pp. 523–530. `doi:10.1109/COASE.2017.8256157`.

[13] M. Vistein, D. Deden, R. Glück, S. Schneyer, Automated production of large fibre metal laminate aircraft structure parts, Procedia Manufacturing 38 (2019) pp. 1300–1307. `doi:10.1016/j.promfg.2020.01.160`.

[14] M. Ghallab, D. Nau, P. Traverso, Automated Planning: Theory and Practice, The Morgan Kaufmann Series in Artificial Intelligence, Elsevier Science, 2004, ISBN: 9780080490519.

[15] B. Miloradović, B. Çürüklü, M. Ekström, A. V. Papadopoulos, A genetic algorithm approach to multi-agent mission planning problems, in: Operations Research and Enterprise Systems, Springer International Publishing, Cham, 2020, pp. 109–134. `doi:https://doi.org/10.1007/978-3-030-37584-3_6`.

[16] B. Miloradović, B. Cürüklü, M. Ekström, A. V. Papadopoulos, GMP: A genetic mission planner for heterogeneous multirobot system applications, IEEE Transactions on Cybernetics (2021) pp. 1–12.`doi:10.1109/TCYB.2021.3070913`.

[17] S. M. LaValle, Planning algorithms, Cambridge university press, 2006, ISBN: 9780511546877. `doi:10.1017/CBO9780511546877`.

[18] K. Shoemake, Animating rotation with quaternion curves, in: Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, Association for Computing Machinery, New York, NY, USA, 1985, pp. 245–254. `doi:10.1145/325334.325242`.

[19] J. J. Kuffner, S. M. LaValle, RRT-connect: An efficient approach to single-query path planning, in: IEEE International Conference on Robotics and Automation, Vol. 2, 2000, pp. 995–1001. `doi:10.1109/ROBOT.2000.844730`.

[20] W. Khalil, E. Dombre, Modeling, Identification and Control of Robots, Butterworth-Heinemann, Oxford, 2002, ISBN: 978-1-903996-66-9. `doi:10.1016/B978-1-903996-66-9.X5000-3`.

[21] D. Engström, H. Hyll, J. Fredsdotter, R. Larsson, Armering i byggprocessen – effektivisering av informationshanteringen, Tech. rep., NCC Teknik, (Date accessed: 29/09/2021) (2011).
URL `http://www.armera.se/pdfer/slutrapport.pdf`

[22] D. Castro-Lacouture, M. Skibniewski, Implementing a B2B e-work system to the approval process of rebar design and estimation, Journal of Computing in Civil Engineering 20 (2006) pp. 28–37. `doi:0.1061/(ASCE)0887-3801(2006)20:1(28)`.

[23] D. Castro-Lacouture, B2B e-work intranet solution design for rebar supply interactions, Ph.D. thesis, Purdue University, (Date accessed: 29/09/2021) (2003).
URL `https://docs.lib.purdue.edu/dissertations/AAI3113780/`