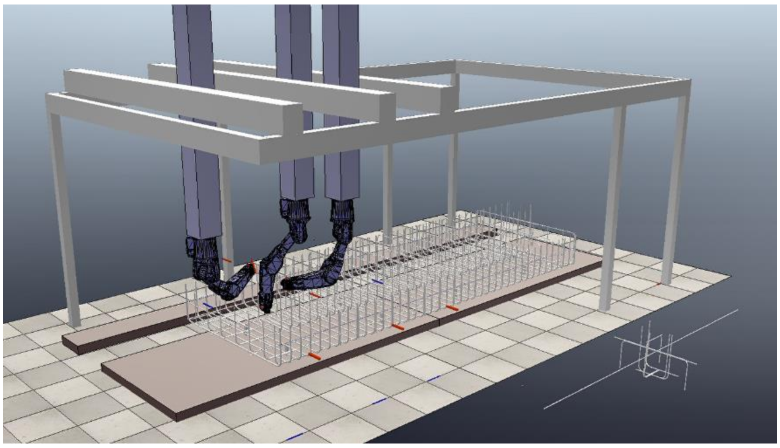
ID: 13666

BYGGAUTOMATION

Banplanering och demonstrator för robottillverkade armeringskorgar



Johan Relefors, Robotdalen AB

2021-02-28

SBUF stödjer
forskning & utveckling
som leder till
praktisk handling

FÖRORD

Projektet har genomförts i samarbete mellan Skanska Sverige AB och Robotdalen AB. Den ursprungliga omfattningen med såväl teoretiskt arbete med banplanering som fysisk demonstration reducerades efter överenskommelse med SBUF till att enbart omfatta den teoretiska delen.

Lars Pettersson, Ulf Håkansson och Abubakar Kathry har deltagit i projektet från Skanska Sveriges sida. Från Robotdalens sida har Johan Relefors deltagit.

Johan Relefors är huvudförfattare till denna rapport. Lars Pettersson har redigerat rapporten och översatt den till svenska.

Projektteamet vill tacka SBUF, Skanska och Robotdalen för finansiellt såväl som tekniskt stöd under projekttiden.

Februari 2021

Johan Relefors

SAMMANFATTNING

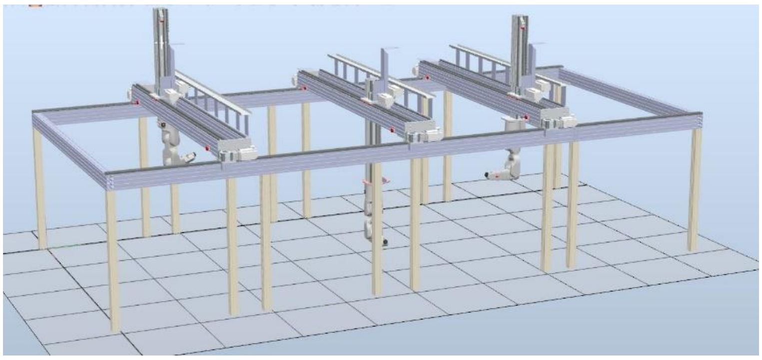
Denna rapport beskriver principerna för algoritmer avsedda för beräkning av de robotrörelser som krävs för att montera samma en armeringskorg beskriven med hjälp av en digital modell. Algoritmerna är avsedda för ett gantry-robot-system bestående av tre industrirobotar hängande i en ställning (”gantry-struktur”) som medger att robotarna kan flyttas såväl i horisontal- som i vertikalled.

Simuleringar gjorda med programvaran CoppeliaSim visar att konceptet, under förutsättning av ideala förhållanden, fungerar som tänkt.

INNEHÅLLSFÖRTECKNING

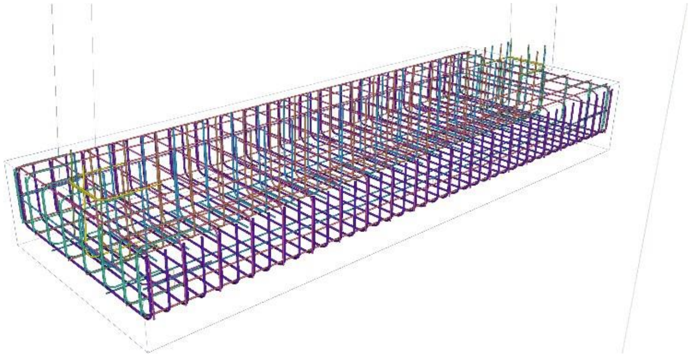
1. INLEDNING

Armering installeras i dagsläget manuellt och ofta järn för järn. Den frågeställning som har varit utgångspunkten för detta projekt är om en maskin skulle kunna utföra detta arbete. Den tänkta maskinen består av tre industrirobotar som hänger upp och ner i en ställning, utformad så att robotarna kan flytta sig i såväl horisontalplanet som i vertikalled (se figur 1). Tanken med maskinen/produktionscellen är att den ska kunna förtillverka armeringskorgar (ref. [1]).



Figur 1. Digital modell av den tänkta maskinen för förtillverkning av armeringskorgar.

Armering dimensioneras och utformas för den konstruktion som de ingår i. Detta innebär att arbetet med att installera armering inte kan liknas vid serietillverkning, förutsättningarna ändrar sig från konstruktion till konstruktion. Att programmera robotarnas rörelser för att installera armeringen för varje enskild konstruktion tar för lång tid. Istället måste armeringsjärnens banor och motsvarande robotrörelser beräknas. Indata för en sådan beräkning måste utgöras av uppgifter om hur armeringen är tänkt att se ut i den färdiga konstruktionen. Det ligger nära tillhands att använda en digital tvilling (se figur 2) av armeringen i den färdiga konstruktionen som indata.



Figur 2. Digital modell av en armeringskorg avsedd för bottenplattan i ett brostöd.

Syftet med detta projekt har varit att studera principerna för sådana beräkningar under ideala förhållanden.

2. BANPLANERING

Allmänt

Den bana som åsyftas i detta projekt är det enskilda armeringsjärnets bana. Järnet ska kunna hämtas i hämta-positionen och sedan förflyttas till slutpositionen utan att kollidera med andra armeringsjärn eller andra hinder. Robotarna som håller i järnet ska kunna förflytta sig längs samma bana och på samma sätt göra detta utan att kollidera med vare sig tidigare monterad armering eller andra hinder. Detta innebär således att hänsyn måste tas till hinder som successivt blir fler allteftersom armeringskorgen färdigställs.

Arbetet med att beräkna det enskilda armeringsjärnets bana från en så kallad hämta-position till den avsedda positionen i konstruktionen och robotarnas rörelser för att skapa denna bana samtidigt som kollisioner undviks benämns banplanering.

I detta projekt har banplaneringen genomförts i följande steg:

1. Skapa den bana längs vilken armeringsjärnet ska röra sig.
2. Skapa robotkonfigurationer för placering av armeringsjärnet i korgen.
3. Skapa robotkonfigurationer för att naja medan en eller flera robotar fortfarande håller i armeringsjärnet.
4. Skapa robotbanorna för att hämta respektive placera armeringsjärnet.
5. Skapa robotbanor för najning och flytta om nödvändigt de robotar som inte najar ur vägen.

Att skapa den bana längs vilken armeringsjärnet ska röra sig

Det enskilda armeringsjärnets bana beräknas med början i järnets position i korgen och slutar där järnet ska hämtas, hämta-positionen. Anledningen till att banan bestäms baklänges är att detta är ett enklare problem att lösa än att placera järnet i korgen. Anledningen till detta i sin tur är i första hand de många möjliga kollisionspunkter som finns i korgen.

Att skapa en bana som beskriver hur ett armeringsjärn flyttas bort från armeringskorgen görs i ett antal bestämda steg. Järnets position sparas efter varje steg. Detta leder till en bana som består av så kallade vägpunkter. Stegen för att åstadkomma detta är:

1. Tillämpa analogin med en repulsiv potential mellan det järn för vilket banan ska beräknas och dess grannar i korgen.
2. Lyft järnet rakt upp så mycket att det kan flyttas fritt ovanför korgen.
3. Flytta armeringsjärnet till korgens mitt.
4. Flytta och rotera armeringsjärnet så att det placeras ovanför hämta-positionen.
5. Sänk ner armeringsjärnet till hämta-positionen.

Att tillämpa en repulsiv potential mellan det järn som ska flyttas och de andra armeringsjärnen i korgen betyder att järnet som ska flyttas skjuts iväg från sina grannar i en riktning där inga kollisioner inträffar. Idén med denna metodik är att finna en väg till positionen i korgen när järnet ska placeras i denna.

Potentialen har valts utgående från dels de initiala respektive önskvärda avstånden mellan det järn som ska flyttas och de andra järnen i korgen, dels det antal iterationer som krävs för att ta bort järnet.

Efter detta steg lyfter vi järnet rakt upp samtidigt som kontroll görs av tänkbara kollisioner. Järnet lyfts till en nivå som medger att järnet kan flyttas utan hinder. Järnet flyttas sedan till en position rakt ovanför den position där det ska hämtas och sänks därefter rakt ned. Under förflyttningen orienteras också järnet så att det har den tänkta orienteringen i positionen där det ska hämtas.

Att placera armeringsjärnet i sin slutposition

Att bestämma lämpliga robotkonfigurationer för placering av järnet görs genom att slumpmässigt prova olika gantry-positioner och använda inverterad kinematik för de resterande 6 frihetsgraderna. Vi kontrollerar sedan att roboten kan släppa järnet i dess tänkta position utan att stöta mot något hinder. Om en robot inte används placeras den i en position som inte kommer att påverka de rörelser som krävs för att placera järnet.

Om najningen inte ytterligare komplicerade uppgiften att placera järnet så vore alla kollisionsfria konfigurationer likvärdiga. Vi behöver dock hålla de robotar som placerar järnet stilla när vi najar, åtminstone för några av najningspunkterna. Därför måste vi anpassa placeringskonfigurationen för att ge plats för najningen. För att göra detta provar vi ett antal placeringskonfigurationer och väljer den med störst skillnad längs gantryns längsta axel.

Att skapa najningspositioner

Så som påpekades i det föregående avsnittet håller vi robotarna som håller i järnet stilla när vi najar, åtminstone för några najningspunkter. Några av najningspunkterna är eventuellt inte möjliga att nå på detta sätt.

Så länge som vi håller de robotarna som håller i järnet stilla provar vi najningspositioner på samma sätt som vi provar placeringspositioner. Vi kontrollerar också att roboten kan utföra en linjär rörelse för att närma sig najningspunkten.

I de fall då vi måste flytta någon av robotarna som håller i järnet så placeras den i en lämplig position där den inte inkräktar på najningsoperationen.

Att skapa banor för att hämta och placera

I detta steg används armeringsjärnets vägpunkter och konfigurationen för att placera järnet i sin slutposition vilket är konfigurationen i den första vägpunkten. Vi provar därefter hela systemet i nästa vägpunkt genom att prova med gantryn och att använda inverterad kinematik för de resterande 6 frihetsgraderna (inverterad kinematik används i detta steg för att kontrollera att robotarnas räckvidd är tillräcklig).

Interpolering för armeringsjärnets orientering görs med sfärisk linjär interpolation och med linjär interpolation för dess läge. Interpolering för robotarna görs med linjär interpolation för gantryn medan inverterad kinematik används för de resterande 6 frihetsgraderna för att möjliggöra att greppen av armeringsjärnen kan följas och att banorna för robotarnas leder måste vara kontinuerliga. I varje steg i interpoleringen utförs också kollisionskontroller för att säkerställa att banan är kollisionsfri.

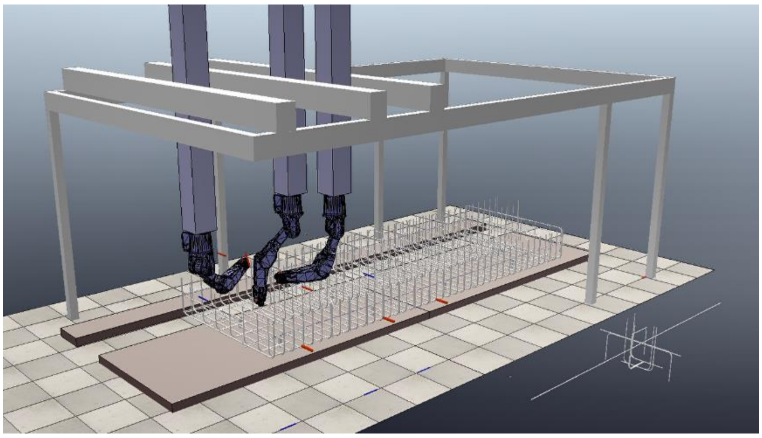
Så snart en bana mellan den första och andra vägpunkten har bestämts fortsätter processen mellan den andra och tredje vägpunkten. Den beräknade konfigurationen vid vägpunkt 2 används som utgångspunkt för robotarna. Processen fungerar inte för varje möjlig slumpmässigt valt prov vid varje vägpunkt. Därför görs flera försök för att hitta en bana. När vi gör detta används en djup-först-sökning där startkonfigurationen vid varje given vägpunkt används ett maximerat antal gånger. Vi avslutar sökningen när en (1) bana till upphämtningsplatsen har identifierats.

Att skapa robotbanor för najning

Najningsbanorna beräknas genom att använda så kallad bi-directional RRT-Connect (BiRRT). Najningsroboten startar i en position där den inte är i vägen för placeringen av armeringsjärnet. Därefter beräknas banan till den första najningspunkten, den andra osv. Om någon av robotarna som används för att placera järnet måste flyttas för att kunna fortsätta beräknas banan för denna rörelse också med hjälp av BiRRT. När detta görs hålls de andra två robotarna stilla.

3. SIMULERING OCH RESULTAT

För att prova de algoritmer som beskrivs ovan har programvaran CoppeliaSim använts. I CoppeliaSim har produktionscellens gantry och robotar modellerats. De robotar som används är ABB:s IRB1200-7/0.7. Figur 3 visar produktionscellen i CoppeliaSim vid installation av armeringsjärn.



De simuleringar med installation av armeringsjärn i en armeringskorg med successivt allt fler armeringsjärn installerade visar att det valda konceptet, under förutsättning av ideala förhållanden, fungerar som tänkt.

REFERENSER

[1] J. Relefors, M. Momeni Kelageri, L. Pettersson, E. Hellström, A. Thunell, A. V. Papadopoulos, T. Nolte, *Towards automated installation of reinforcement using industrial robots*, in: The 24th IEEE Conference on Emerging Technologies and Factory Automation, 2019, pp. 1595–1598.

URL http://www.es.mdh.se/publications/5704-

Towards Automated Installation of Reinforcement Using Industrial Robots

Johan Relefors[1], Mahdi Momeni[2], Lars Pettersson[2,3], Erik Hellström[1],
Anders Thunell[1], Alessandro V. Papadopoulos[2], and Thomas Nolte[2]
[1]Robotdalen AB, Västerås, Sweden
[2]Mälardalen University, Västerås, Sweden
[3]Skanska Sweden AB, Stockholm, Sweden

Abstract—**The construction industry is today among the least automated industries with a long tradition of utilizing manual labour. Despite the potential benefits of automation, only a few examples of using robots to automate (parts of) construction have been presented over the past years. In this paper we present our ongoing work towards automated installation of reinforcement, a traditionally very heavy and labour intensive work. We use industrial robots and we discuss the potential benefits and challenges of such robotic automation in construction. Our overall goal is to achieve a fully automated robotic solution for flexible serial production of custom made non-identical reinforcement cages. In the paper we highlight and analyse the main challenges that must be addressed in order to reach a functioning and efficient solution.**

I. INTRODUCTION

Employment and application of robots in various production tasks has in recent years gained increasing interest in manufacturing industries. The number of robots deployed worldwide is estimated to increase to about 2.6 million units by 2019. About 70% of them will be used in the automotive, electrical/electronics, and metal and machinery industries[1]. There are also considerable research and development efforts in this field. While there are initiatives towards automation in the construction domain both in research [1]–[4], and in industry, e.g., Built robotics[2], or MX3D[3], the overall trend has not been reflected in the construction industry.

One key difference between the construction industry and other industries with higher degrees of automation, like manufacturing industries where robots have been extensively employed, is that construction projects are typically one of a kind. This means that less time can be put into product-specific automation and more time is spent on changeovers. Furthermore, each structure is unique in its details, thus automated solutions require special and careful considerations.

While construction projects present challenges for automation, the gains are potentially large [4]. Moreover, work in the construction industry is often strenuous and happens under hazardous and harsh conditions. This is reflected in a high risk for work accidents. The above-mentioned, as well as other issues, create a huge potential need for utilizing automation techniques in the construction industry, with a promise for increased safety and efficiency, as well as reduced costs. In this work, we address automation of rebar cage (reinforcement) fabrication, looking in particular towards civil engineering structures. In particular, we consider three industrial robots hanging on a gantry for the installation of rebars, see Fig. 1. Since each and all rebar cages are unique in their details and are designed for a particular structure, we aim for flexible serial production. Therefore, the challenge is to automate the process that goes from the design of the 3D digital (CAD) model to the construction of the reinforcement cage, exploiting the described robotic system. Such process presents several challenges, described in this paper, ranging from the task and path planning for the robots, and the management of the flexible moving parts, and the coordination of the multi-robot system. In this paper, we describe and analyze these challenges, and we outline our research directions towards to address them.

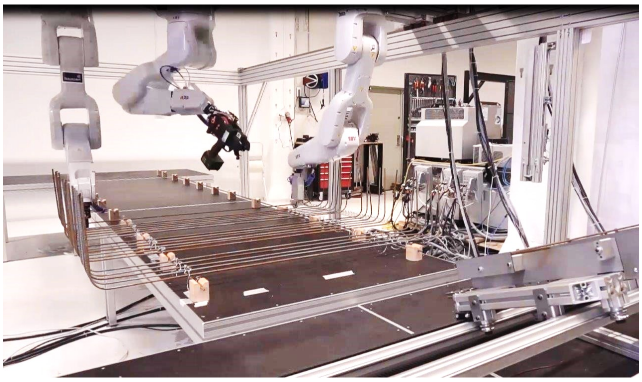


Fig. 1. Down-scaled system.

II. PROBLEM STATEMENT AND MISSION

The aim of this work is to answer the question: *Is it possible to fabricate reinforcement rebar cages using industrial robots?* The question is interesting for many reasons. First, prefabricating rebar cages, regardless of method, has the potential to shorten build times. Second, is that automation would

---

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

[1] https://ifr.org/

[2] https://www.builtrobotics.com/

[3] https://mx3d.com/

978-1-7281-0303-7/19/$31.00 ©2019 IEEE

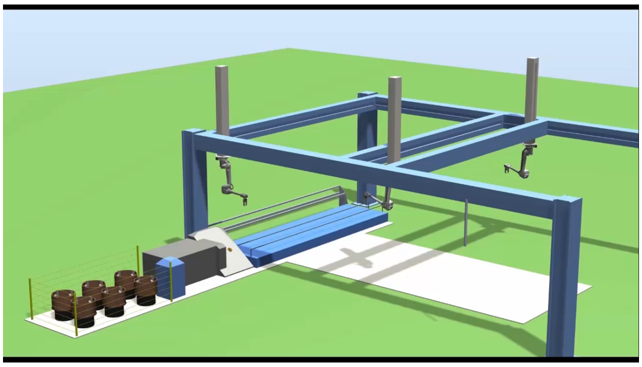


Fig. 2. Gantry robot systems simulation.

reduce the amount of heavy and hazardous manual work that current methods of installing reinforcement lead to. Finally, it is an interesting question to answer from an automation and production perspective. We started using simulations set up in ABB RobotStudio, along with experiments in the lab. In RobotStudio, the construction of a rebar cage used in bridge construction was simulated. We also simulated the construction of a simplified down-scaled rebar cage, which was later built in a lab setting (see Fig. 1). In the lab we used three ABB IRB1200 robots hanging upside down, mounted on a gantry robot with two degrees of freedom for each robot (see Fig. 2). We did this both by manually programming the gantry robot paths and by generating its paths from rebar and tie point coordinates.

To be able to build the rebar cages in the lab, tools had to be developed. A special tool for rotating a rebar without moving the robot is used in this setting, and it simplifies the programming of the robot paths. A tool for tying rebars together was also constructed, by adaption of a manual tie gun. The tying tool was not good enough for production.

In this paper, we aim at building rebar cages using robots hanging from the gantry further, based on our experience from the previous set-up. A goal that we have set is to develop software which takes a CAD-model of a rebar cage as input, and produces robot paths for building the rebar cage, using the gantry-robot system, as output, see Fig. 3. We also want to test the generated paths on a physical system. The reason that we need to generate the robot paths, instead of programming them by hand, is that rebar cages used in civil engineering and building structures are unique. The reason that each case is unique is that they are designed to the precise dimensions and the precise loads at a given location in the structure.

III. CHALLENGES

There are several challenges that need to be tackled for the automated construction of rebar cages. The problem can be split into four major challenges: (i) rebar installation order, (ii) path planning for placing rebars, (iii) path planning for tying rebars, and (iv) transitioning to the real world.

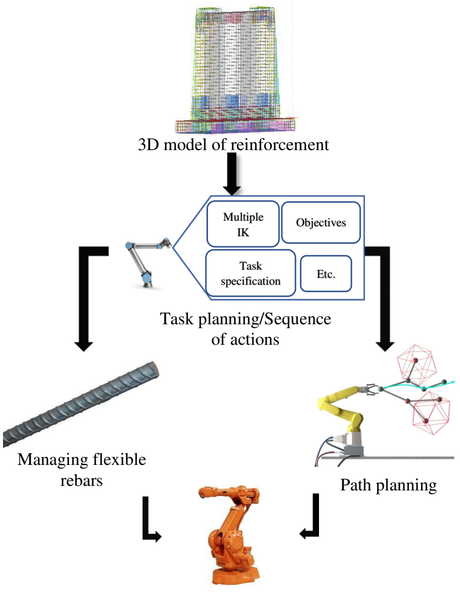


Fig. 3. Overall process scheme.

A. *Rebar installation order*

The first challenge concerns the ordering in which the rebars must be placed, i.e., how to generate a valid order based on a CAD-model of the final rebar cage. Finding such an order is a challenging problem given the fact that there are often more than 100 rebars to place, meaning that a pure combinatorial search is not possible. Task planning approaches will be considered, e.g., [5], focusing on heuristics able to provide sub-optimal yet feasible solutions.

For example, the ordering problem can be addressed by observing that the positions where rebars are to be placed are most likely often more constrained than the positions where the rebars must be picked. Backward chaining approaches can be helpful in addressing such complexity [6]. More specifically, given the target 3D CAD-model of the fully constructed cage we determine where to start, and we iteratively compute the gantry robot motions to remove all the rebars to disassemble completely the structure. As noted above, this leads to a very large search space. Ideally, we would like to disqualify all of the non-working orderings without computing any actual robot motions. If that is not possible we would at least like to limit the actual robot motions that we need to compute. One way of lowering the number of robot motion computations needed is to apply some heuristics to disqualify certain sequences. For example, consider a rebar which can move on its own. If, in this simplified problem, the rebar cannot be removed from the cage then it will not be possible when adding the complications of the gantry, robots and tools. One way of implementing this idea is to apply a repulsive potential between the rebar which we are currently trying to remove and all other rebars, similarly to what is done with obstacle avoidance in potential field methods [7]. If, as a result of the force from the potentials, the rebar moves away from all other rebars with some clearance, then it might be possible to remove using a robot. Otherwise, we can safely say that it cannot be removed yet. Another idea to reduce the number of possible orderings is to try to introduce precedence constraints among the rebars [5]. The basic idea is then that some rebars cannot be attached to the cage before others are already in place. A wrong ordering in this case could mean that a rebar is floating freely, not attached to anything, or that there are not enough tie points available given the current cage to tie the given rebar into place.

By using ideas like the ones above, the ordering can be solved as a separate problem. However, being able to remove the rebar from the cage and finding that it can be attached to the cage is only part of the problem, the rebar must also be tied to the cage. This step will affect the possible valid orderings. Again, we might not need to try to compute the full tie path to disqualify an ordering. It would be enough to show, for example, that no robot can reach the tie position.

To finish this section we point out that not every rebar cage is possible to build using the setup we have in mind. A human might temporarily bend a rebar ever so slightly to fit it over some structure, something which is more difficult to achieve with robots. Another example is that human are very good at re-grasping whereas when a robot holds a rebar in a well defined grip it is most likely not a good idea to let go of that grip. The reason being that any re-grasping will most likely not produce the same level of precision as grasping in a fixture, where the piece to be picked up is fixed in a well defined position. Another thing humans do at a construction site is to adapot to incomplete drawing. Rebar cage drawings are most often incomplete in the sense that the people building the cages are expected to add (temporary/support) rebars that will make the structure possible to build.

B. Path planning for placing rebars

In this section, we assume that the rebars, the gantry and the robots are completely stiff, that is we completely ignore any deflection effects. The problem is then to pick up a given rebar, using given gripping points, from a fixture using one or more of the robots, move the rebar to its final position in the rebar cage and finally releasing the rebar and moving the robots away. In doing this, no collision is allowed. We assume that the rebar which is to be placed is part of a valid ordering. Also note that we do not consider the challenge of tying the placed rebar to the existing cage here, treated next.

When generating paths for the robots to solve a problem like this, where we can assume that the fixture is easily accessible and the final position is more difficult to reach, it is possible to still reason in backward chaining fashion. We can consider the case where the robot is supposed to remove the rebar from its position in the cage and place the rebar in the fixture. Given this choice we expect that removing the rebar from the cage is the difficult part of the problem and we will therefore limit the discussion to removing the rebar away from the cage.

There are several possible strategies to try to solve the problem of removing the rebar from the cage. We are currently working along the lines of: (i) computing collision free robot configurations where the gripping robots are holding the rebar which is to be removed, or, (ii) computing a path for the system of robots and rebar which removes the rebar from the cage. Since we assume that it is possible to place the rebar in the rebar cage the first step should be fairly easy. It amounts to compute the inverse kinematics for the robot-gantry system with the constraint that there can be no collisions. Since each gantry robot system has 9 degrees of freedom (DOFs), methods for resolving the the robot redundancies are needed [8]. Alternatively, it is possible to place the gantry in a given location and then compute the inverse kinematics for the robot manipulator. Both of these methods can generate multiple possible robot configurations for each gantry-robot system. To find collision free paths, the possible configurations for each gantry-robot system can be tested for collision against each other until a collision free configuration for the full system is found. Collision-free motion can be included in the redundancy resolution of the robot, to reduce the number of possible configurations.

The second step, to compute a path for the system to remove the rebar from the cage, is equivalent to the extensively studied problem of robot path planning. One often used class of methods are randomness based methods based on the RRT/RRT* algorithm [9]–[11]. RRT/RRT* efficiently explore configuration space even for high dimensional problems. In our case we have a system consisting of 3 gantry-robot systems, each with 9 DOFs giving the full system 27 DOFs. To simplify the planning we note that in cases where one or more robots are not used, the computed plan only needs to keep the robot out of the way. Furthermore, when more than one robot holds the rebar there are constraints on the relative movements of the gantry-robot systems. For example, given that two robots are gripping the rebar in a fixed way (a grip which establishes a rigid transformation between the tool and the rebar) the number of DOFs is reduced from 18 to 12.

C. Tying the rebars

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 standard tying tool which has been adapted for robotic applications. The quality of the knots produced by this tying tool is probably not high enough and we are investigating alternatives. One possibility would be to make a combined gripping and tying tool. The challenge presented here is restricted to the assumption that there is no deflection, leaving real world considerations to future work.

While the tying is being performed the rebar must be held in place. Although it is possible to imagine some temporary means of holding the rebar in place while tying, we believe that at least one of the robots not involved in the tying can hold the rebar. When it comes to plan the actual robot motion for tying, there are many alternatives for this kind of path planning, e.g., RRT-like methods. If possible, the robots not involved in the tying will be standing still, making the problem into a 9 DOFs planning problem.

D. Transition to real world

In the previous sections, we considered the problem from a purely theoretical point of view, neglecting real world effects such as deflections and misalignments. These effects can be treated in at least two different ways: (i) calibrate the system to get the real world system behavior closer to the idealized situation, or (ii) adapt the algorithms to the real world behavior. Most likely a combination of the two will give best performance.

We tested the transition from the simulation to the actual gantry-robot system setup in a lab, with hard-coded robot paths. In this setup, we did use both calibration and adaption. The effects which we calibrated for were deflection in the gantry construction and misalignments in the mounting of the robots and in the gantry.

To calibrate for misalignments, we measured the position of a prism fastened to the robot TCP using an external measurement tool in a few different configurations and calculated the actual robot base position and orientation. We then adjusted the orientation and position of the robot base in the controller accordingly.

To calibrate for deflection in the gantry, we let the robots assume a position where its center of gravity is fairly well aligned with the base. We then made position measurements using an external measurement tool. This gives an offset between the actual position and the theoretical one. Therefore, we moved the robot position using the gantry producing a grid of such measurements giving a map of offsets. This map was used to produce an offset for each point in the robot program. This is not entirely unproblematic since if the robot is close to its limits such a displacement might make the point out of reach. Moving a point can also change which robot configurations can reach the point. These two effects will have to be taken into account when generating robot movements, meaning that we are adapting our algorithms to reality.

Another effect on the gantry comes from the torsion on the gantry when the robot is working in a position where it stretches to the side. This torsion means that the robot TCP will depend on where the robot is working and what weight it is currently lifting. We have measured this in the lab and found the effect to be small enough to neglect. This might not be the case when producing real size rebar cages however.

An effect that cannot be calibrated away is the deflection of rebars. This effect enters both in placement and tying of rebars. When placing rebars, collisions that would not have occurred for completely stiff rebars might occur when the rebars deflect. This might also happen if an angle of a rebar is not precisely as intended.

When tying rebars together deflection often means that the point in the rebar which is supposed to be tied to the cage is not at the location where it should be. The most obvious example being that the point is below where it should. However, this is not the only direction a point may be displaced. This can happen when a rebar is lifted with one tool completely fixing the rebar while another is letting the rebar rotate and slide along the rebars longitudinal direction. The deflection of the rebar can then mean that the longitudinal position of the point is shifted as well. The only location on such a rebar which is reliable is the gripping point of the fixating tool. To address this problem, it might be possible to start tying close enough to the fixating tool that the precision becomes good enough. Alternatively, it might be possible to displace the rebar position such that the tying location on the rebar is in the correct position. This would require accurate simulation of rebar deflection. Finally, it is possible to use some kind of temporary support structure.

So far, we have mainly discussed physical effects as well as ways of integrating them into the motion planning. We have most likely missed some effects and also the solutions we have proposed have finite accuracy. This leads to the topic of tolerances. When testing a system like this in the real world there will have to be tolerances which allow for the modifications that we propose as well as for the lingering errors. This, together with the modifications themselves, means that a rebar cage that is possible to build in theory might not be possible to build in practice.

IV. SUMMARY AND WORK-IN-PROGRESS

In this paper, we presented our ongoing work towards robotic construction of rebar cages. As a first step towards a solution, we are considering rebar placement in an ideal, simulated, world with no deflection. We will then move on to develop techniques for computing rebar installation ordering and paths for tying of rebars to the cage. Once completed, we are ready to start simulating deflection and to reiterate the steps above with this new source of difficulty. Finally, we will perform real world tests, first using small rebar cages in a lab setting (see Fig. 1) and then full scale tests.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the team at Robotdalen, both current and previous members, especially Ingemar, Hugo and Josef.

REFERENCES

[1] B. Dolinšek and J. Duhovnik, “Robotic assembly of rebar cages for beams and columns,” *Automation in Construction*, vol. 8, no. 2, pp. 195–207, 1998.

[2] A. Mirjan *et al.*, *Building a Bridge with Flying Robots*. Springer, 2016.

[3] H. Ardiny *et al.*, “Are autonomous mobile robots able to take over construction? a review,” *Int. J. Robotics*, vol. 4, no. 3, pp. 10–21, 2015.

[4] R. Bogue, “What are the prospects for robots in the construction industry?” *Industrial Robot*, vol. 45, no. 1, pp. 1–6, 2018.

[5] B. Miloradović *et al.*, “Extended colored traveling salesperson for modeling multi-agent mission planning problems,” in *ICORES*, 2019, pp. 237–244.

[6] R. Poli and W. B. Langdon, “Backward-chaining evolutionary algorithms,” *Artificial Intelligence*, vol. 170, no. 11, pp. 953–982, 2006.

[7] S. Ge and Y. Cui, “Dynamic motion planning for mobile robots using potential field method,” *Auton. Rob.*, vol. 13, no. 3, pp. 207–222, 2002.

[8] A. Reiter *et al.*, “On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators,” *IEEE Trans. Ind. Inf.*, vol. 14, no. 4, pp. 1681–1690, 2018.

[9] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” Tech. Rep., 1998.

[10] S. M. LaValle and J. J. Kuffner Jr., “Randomized kinodynamic planning,” *Int. J. of Rob. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[11] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *Int. J. of Rob. Res.*, vol. 30, no. 7, pp. 846–894, 2011.