

Optimization of Trajectories for Cable Robots on Automated Construction Sites

Roland Boumann, Patrik Lemmen, Robin Heidel and Tobias Bruckmann

University of Duisburg-Essen, Germany

E-mail: {roland.boumann, patrik.lemmen, robin.heidel, tobias.bruckmann}@uni-due.de

Abstract -

In contrast to conventional serial robot arms, a cable robot offers special characteristics which make it predestined for usage in automated construction: Cable robots use a robust and simple mechanical layout, including a frame, motor-driven winches, pulleys and cables connected to the end effector in a parallel kinematic structure. Derived from BIM data, suitable trajectories need to be generated for every payload, e.g. as set values for the robot control. This paper focuses on the generation and optimization of such trajectories, considering several requirements and optimization criteria, using a hybrid particle swarm algorithm for global optimization. The optimization costs include transportation time, cable forces, collision avoidance, stiffness and movements of the pulleys. Some of these criteria are in conflict of aims, e.g. a short transportation time and low cable forces, which is resolved by weights in the cost function.

Keywords -

Cable Robot; Optimization; Path Planning; Construction

1 Introduction and State of Art

Over the last decades, repeatedly, the automation of bricking was considered. Projects in the 1990s, such as ROCCO, with a payload of around 500 kg and a range of up to 8.5 m [1], and BRONCO [2], use modern and economic approaches, using conventional industrial robots, partly on mobile platforms.

At ETH Zurich, mobile platforms have also been equipped with robot arms to investigate automated construction processes, initiated by Gramazio and Kohler in 2011. Different sensor concepts and tools were tested in installations such as “Endless Wall”, “Stratifications” and “Fragile Structure” [3, 4].

Currently, the Australian company Fastbrick Robotics is developing large manipulators called Hadrian and HadrianX, which are designed to bring building blocks along a large manipulator arm via belt conveyors to a desired position on the building site, using special bricks.

In parallel, the American company Construction Robotics LLC offers the SAM (Semi-Automated Mason). SAM employs a conventional serial robot on a movable

platform or mounted on a large manipulator to achieve all the necessary poses on the construction site. According to Brehm [5], SAM is designed as a semi-automated system that is able to increase the productivity of a bricklayer by a factor of five, whereby the bricklayer is still responsible for creating the calibrating layer, the joints and setting the corners. SAM works with small-format bricks with grouted butt joints, which are applied using the dipping method.

2 Concepts for Bricking using Cable Robots

To construct a building by a robot, the size of the required workspace is supposed to be one of the dominant technical challenges. Here, cable robots offer outstanding advantages in terms of workspace size, stiffness, modularity and mobility [6, 7, 8]. Currently, first feasibility studies on cable robots for application in bricking have been done in lab scale. To extend this to practical full scale experiments, recently projects have been initiated that face specific technical challenges for the prototype, which are discussed here in short.

To allow for fast assembly and dismantling, and easy adaption to variable dimensions of the building to be constructed, the frame of the cable robot providing suspension of the end effector must consist of lightweight modular elements. Still, stiffness and constructive accuracy should not be less than with a fabricated steel construction. Here, good calibration procedures for setting up the geometric parameters (e.g. pulley positions) are needed in the future. Furthermore, the lower cable deflection pulleys – see Figure 1 – must be vertically movable in order to avoid collisions between cable and building during the construction process with a growing building structure. For good model accuracy – and thus for good positioning accuracy of the system – the guides of the deflection pulleys must be correspondingly precise.

Furthermore, the robot’s end effector – realized as a platform connected to the cables, carrying the brick gripper and a set of sensors – needs a home position, e.g. for revision purposes. As the ground is blocked by the building, it is necessary to construct a set-down position on the upper frame structure.

Another component specifically to be developed for this application is the brick gripper. For the exact positioning

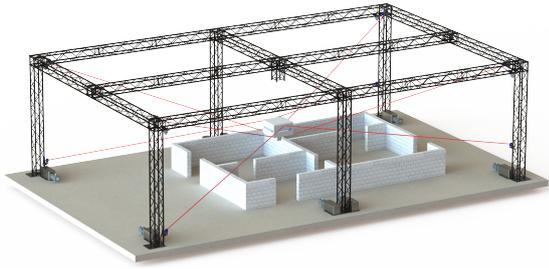


Figure 1. Principal CAD drawing of a cable robot for automated construction sites

of the brick, the brick must be gripped in a precisely defined way in order to guarantee a defined brick placing process. A sophisticated sensor concept for detecting both the stone to be gripped, as well as the objects and bricks already placed, is essential. In addition, the stone gripper must generate the gripping force passively to avoid the brick falling down in the event of power loss. Still, a form-fit gripping of the stone is not possible when using conventional bricks and mortar to be applied to the brick.

The use of a cable robot on a site also places special demands on the protection class of any components. The robot must be designed to be protected against dust and dirt and the influence of different weather conditions. The safety concept for working on a construction site must also be reviewed, which may require the design of new functions, such as the self-locking gripper.

3 Methodology

While the realization of the prototype components is a complex mechatronic task, another open issue is the model-based optimization of the robot and its behaviour. In this work and as a first step, the motion of the end-effector as well as the motion of the lower four cable deflection pulleys is optimized for the bricking process. This is resolved by a comprehensive physical model discussed in this work, and the careful formulation of a numerical optimization problem.

4 Numerical Optimization Approach

To be able to optimize the cable robot behaviour for bricking, its physics and properties will be modeled in the next sections.

The numerical optimizers to be used base on defining a vector of scalar parameters to be optimized, their boundaries and a scalar cost function. Thus, the properties to be optimized need to be modeled and evaluated resulting in a scalar cost function for each property. The formulation of those scalar cost functions will be given in the following.

Noteworthy, the different cost functions, each representing a property to be optimized, need to be merged. In this work, the multiobjective optimization is resolved by simply adding the single cost values \mathcal{V}_χ using weights \mathcal{W}_χ where all properties changing along a robot motion are discretized in n_s time steps, see section 5.

Some properties are subject to physical limits. As some of them are not parameters but result of the models to be developed, the question of considering those boundaries arises. Thus, this work uses penalty terms \mathcal{P}_χ .

Accordingly, the total cost $\mathcal{V}_{\text{total}}$ is computed by summarizing the costs and penalties for all properties χ over all n_s time steps .

5 Trajectory Modeling

The trajectory describes the motion of the robot as a path defined over time. Accordingly, it covers information on pose, velocity and acceleration. Hence, a model for the description of these quantities must be found that allows a parametrization by the numerical optimizer. Here, splines – i.e. a composition of interconnected polynomials – are a popular choice. Within this work, only the platform position \mathbf{r}_P with three degrees-of-freedom (DOF) as well as the lower four pulley positions \mathbf{s} may change over time and thus are described by splines. As shown in table 1, for the platform position, six parameters per DOF are added to the optimization variables, resulting in 18 parameters for the description of the translational movement in three DOF. Note, rotations are not performed. Additionally, per segment a time duration $t_k, k = 1, 2, 3$, is defined as an optimization parameter, which adds three parameters. Resulting, the vector of optimization parameters contains $18 + 3 = 21$ elements subject to numerical optimization.

Four parameters per pulley are added to the optimization variables, resulting in 16 parameters for the description of the movement of the four lower pulleys. The segment times are copied from the platform motion. Summarizing, $21 + 16 = 37$ parameters are used in the numerical optimizer.

To evaluate the properties of the robot along a motion described by these splines, each spline needs to be discretized. Let $\dot{s}_i(q), i \in \{1, 2, 5, 6\}$, be the velocity of the lower pulleys along the linear axis at the q^{th} discretization step of one spline segment, n_{s_k} be the number of discretization points along one spline segment and n_k the number of spline segments.

While \mathbf{r}_P and \mathbf{s} as well as their temporal derivatives can be efficiently bound at the knots by limits (see table 3), they need to be bound along the whole trajectory as well. Thus, they are checked on each step and penalty terms $\mathcal{P}_{\text{traj}}$ and $\mathcal{P}_{\text{pulleys}}$, respectively, are associated. Noteworthy, as high robot dynamics per se are desirable, but already reflected by time, $\mathcal{V}_{\text{traj}} = \mathcal{W}_{\text{traj}} = 0$.

Table 1. Description of spline and optimizer parameters.

	Start (1 st knot)	2 nd knot	3 rd knot	Goal (4 th knot)
Spline segment	1	2	3	
Platform position; one spline per DOF; i.e. three splines in total				
Spline order	7	7	7	
Constraints	\mathbf{r}_P is given; $\dot{\mathbf{r}}_P, \ddot{\mathbf{r}}_P, \ddot{\mathbf{r}}_P$ are set to zero	$\mathbf{r}_P, \dot{\mathbf{r}}_P, \ddot{\mathbf{r}}_P$ are parameters, $\ddot{\mathbf{r}}_P$ is set to zero	$\mathbf{r}_P, \dot{\mathbf{r}}_P, \ddot{\mathbf{r}}_P$ are parameters, $\ddot{\mathbf{r}}_P$ is set to zero	\mathbf{r}_P is given; $\dot{\mathbf{r}}_P, \ddot{\mathbf{r}}_P, \ddot{\mathbf{r}}_P$ are set to zero
Σ of constraints per DOF	4	4	4	4
Σ of optimization parameters in total	0	3×3	3×3	0
Pulley position; one spline per pulley; four splines in total				
Spline order	3	5	3	
Constraints	s is a parameter; \dot{s}, \ddot{s} are set to zero	s is a parameter; \dot{s}, \ddot{s} are computed from first segment	s is a parameter; \dot{s}, \ddot{s} are computed from last segment	s is a parameter; \dot{s}, \ddot{s} are set to zero
Σ of constraints per pulley	3	1 (1 st segment); 3 (2 nd segment)	1 (last segment); 3 (2 nd segment)	3
Σ of optimization parameters in total	4×1	4×1	4×1	4×1

To avoid collisions (see section 6.3), a reconfiguration of the cables might be necessary. Yet, in terms of power consumption, all movements of the pulleys require energy and are thus associated with costs. A detailed dynamic modeling of the pulley movement is neglected in this paper. For simplicity, viscous friction with a friction constant μ_f is assumed and results in costs for movement of the pulleys, scaled down to one pulley.

$$\mathcal{V}_{\text{pulleys}} = \frac{1}{4} \sum_{k=1}^{n_k} \sum_{q=1}^{n_{s_k}} (\|\dot{s}_1\| + \|\dot{s}_2\| + \|\dot{s}_5\| + \|\dot{s}_6\|) \mu_f \quad (1)$$

As the robot motion should be completed as fast as possible, the segment durations t_k can be simply summarized to the total time t_{total} . The cost function $\mathcal{V}_{\text{time}} = (t_{\text{total}} - n_k t_{\text{min}}) \frac{1}{n_k t_{\text{min}}}$ computes a value with respect to the smallest possible time. Furthermore, it puts this difference in relation to the smallest possible time, which allows an interpretation. The minimum reachable cost value is zero. As all t_k are directly defined as optimization parameters, their limits (see table 3) can be directly considered by the optimizer avoiding penalty terms, i.e. $\mathcal{P}_{\text{time}} = 0$.

6 Modeling

6.1 Kinematics and Dynamics

The cable robot, as shown in Figure 2, is referenced in the inertial coordinate system $\uparrow \mathcal{B}$. The end-effector carries the platform-fixed coordinate-system $\uparrow \mathcal{P}$. The

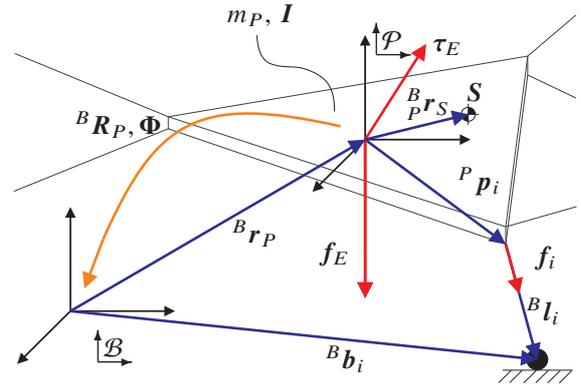


Figure 2. Cable-robot model parameters

robot is driven by m cables and has n degrees-of-freedom and therefore a redundancy of $r = m - n$. The posture in the inertial coordinate-system, consisting of position ${}^B \mathbf{r}_P$ and orientation ϕ of the platform, is ${}^B \mathbf{x}_P = [{}^B \mathbf{r}_P \ \phi]^T$. The rotation matrix ${}^B \mathbf{R}_P$ describes the orientation of the platform with respect to $\uparrow \mathcal{B}$, in the means of roll-pitch-yaw angles. ${}^B \mathbf{l}_i$ are the cable vectors, obtained by the inverse kinematic.

$${}^B \mathbf{l}_i = {}^B \mathbf{b}_i - \underbrace{({}^B \mathbf{r}_P + {}^B \mathbf{R}_P {}^P \mathbf{p}_i)}_{{}^B \mathbf{p}_{B_i}}, \quad 1 \leq i \leq m \quad (2)$$

The cables enter the workspace at the end of the base vectors ${}^B \mathbf{b}_i$, which are derived by using inverse pulley kinematics, see [9]. As introduced, the system can be

reconfigured by moving the pulleys along a straight line in vertical direction. The attachment points of the cables at the platform are ${}^P \mathbf{p}_i$. The cable force vector $\mathbf{f} \in \mathbb{R}^{m \times 1}$ contains all cable forces f_i in direction \mathbf{v}_i of each cable i . Every i^{th} cable exerts the tension f_i on the platform.

$$\mathbf{f}_i = f_i \cdot \frac{\mathbf{l}_i}{\|\mathbf{l}_i\|_2} = f_i \cdot \mathbf{v}_i, \quad 1 \leq i \leq m \quad (3)$$

The static force equilibrium at the platform is given by

$$-\mathbf{w}_E = \begin{bmatrix} -\mathbf{f}_E \\ -\boldsymbol{\tau}_E \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_m \\ \mathbf{p}_1 \times \mathbf{v}_1 & \dots & \mathbf{p}_m \times \mathbf{v}_m \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} = \mathbf{A}^T \mathbf{f}. \quad (4)$$

Herein, \mathbf{A}^T is the structure matrix of the robot and the forces and torques at the platform are included in \mathbf{w} . Setting up Newton-Euler equations [10] in \mathcal{B} , one obtains

$$\begin{aligned} & \underbrace{\begin{bmatrix} m_P \mathbf{E}_3 & -m_P {}^B \mathbf{R}_S \mathcal{H} \\ m_P {}^B \mathbf{R}_S & {}^B \boldsymbol{\theta}_P \mathcal{H} \end{bmatrix}}_{\mathbf{M}(\mathbf{x}_P)} \underbrace{\begin{bmatrix} {}^B \ddot{\mathbf{r}}_P \\ \ddot{\boldsymbol{\phi}} \end{bmatrix}}_{\dot{\mathbf{x}}_P} + \dots \\ & + \underbrace{\begin{bmatrix} m_P [({}^B \dot{\mathcal{H}} \dot{\boldsymbol{\phi}} \times {}^B \mathbf{r}_S + (\mathcal{H} \dot{\boldsymbol{\phi}} \times ({}^B \mathcal{H} \dot{\boldsymbol{\phi}} \times {}^B \mathbf{r}_S))] \\ {}^B \boldsymbol{\theta}_P \dot{\mathcal{H}} \dot{\boldsymbol{\phi}} + (\mathcal{H} \dot{\boldsymbol{\phi}} \times ({}^B \boldsymbol{\theta}_P \mathcal{H} \dot{\boldsymbol{\phi}})] \end{bmatrix}}_{\mathbf{K}(\mathbf{x}_P, \dot{\mathbf{x}}_P)} \dots \\ & \dots + \underbrace{-{}^B \mathbf{w}_E}_{\mathbf{Q}(\mathbf{x}_P, \dot{\mathbf{x}}_P)} = {}^B \mathbf{A}^T \mathbf{f}. \quad (5) \end{aligned}$$

The mass of the platform is m_P and its inertia tensor is ${}^B \boldsymbol{\theta}_P$. \mathbf{E}_3 is a 3×3 identity matrix. Matrix \mathcal{H} and its time derivative $\dot{\mathcal{H}}$ can be obtained from the kinematic Kardan equations [11]. Matrix ${}^B \mathbf{R}_S$ is a transformation matrix between the center of gravity and origin of the platform coordinate system. $\dot{\mathbf{x}}_P$ and $\ddot{\mathbf{x}}_P$ are the first and second time derivative of the end-effector pose \mathbf{x}_P . The vector from the platform coordinate system to its center of gravity is ${}^B \mathbf{r}_S$. $\mathbf{M}(\mathbf{x})$ is the mass matrix of the platform, $\mathbf{K}(\mathbf{x}_P, \dot{\mathbf{x}}_P)$ contains Coriolis and centrifugal forces and torques and $\mathbf{Q}(\mathbf{x}_P, \dot{\mathbf{x}}_P)$ holds all remaining forces and torques including gravitational forces, friction and disturbances.

To obtain set point cable forces for the control, based on a desired trajectory, eq. (5) needs to be solved for \mathbf{f} with a given \mathbf{w} . As the cable can only pull put never push, a minimum tension f_{\min} in the cables is necessary. To avoid cable breaks, the cable forces must not exceed a maximum tension f_{\max} . Several well-known methods exist to solve this problem [12], which differ e.g. in real-time capability or the resulting force level.

As cable forces correspond to the power usage of the system, low cable forces are desired in terms of costs. Let $f(q)$ be the cable force distribution at the q^{th} discretization point of one spline. Per spline, the magnitude of the cable

forces along each spline segment is summarized and then weighted using the time per spline t_k .

$$f_w(k) = \frac{t_k}{t_{\text{total}}} * \frac{1}{n_k} * \sum_{q=1}^{n_{sk}} \sum_{i=1}^m f_i(q) \quad (6)$$

Based on the weighted forces f_w , the average cable force with respect to one cable throughout the whole trajectory is calculated using the total number of discretization points n_s and the number of cables m . Further on, the value gets normalized with respect to f_{\min} to allow for zero costs when the optimal value is reached. The resulting costs for cable forces are

$$\mathcal{V}_{\text{force}} = \left(\sum_{k=1}^{n_k} f_w(k) * \frac{1}{n_s} * \frac{1}{m} - f_{\min} \right) * \frac{1}{f_{\min}}. \quad (7)$$

To cope with invalid cable force distributions in the optimization process, a penalty term is added to $\mathcal{P}_{\text{force}}$ as soon as at one time step, the cable forces are beyond the force limits (see table 3) and thus invalid. To allow the optimizer to iteratively resolve disadvantageous trajectories, each additional time step with invalid forces adds another increment to $\mathcal{P}_{\text{force}}$.

6.2 Stiffness

Stiffness is a critical issue for large and elastic manipulators such as cable robots. Assuming a linear behavior, the stiffness can be described by a spring equation such as

$$\delta \mathbf{w} = \mathbf{K}(\mathbf{x}_P) \delta \mathbf{x}_P, \quad (8)$$

where $\delta \mathbf{w}$ is the reaction force caused by a displacement $\delta \mathbf{x}_P$ that occurs due to a stiffness defined in a diagonal matrix \mathbf{K} , where the latter is determined for a platform position \mathbf{x}_P . $\mathbf{K}(\mathbf{x}_P)$ is computed according to [13, 14] as

$$\mathbf{K}(\mathbf{x}_P) = - \underbrace{\frac{\partial \mathbf{A}^T}{\partial \mathbf{x}_P} \mathbf{f}}_{\mathbf{K}_g} + \underbrace{\mathbf{A}^T \mathbf{K}_l \mathbf{A}}_{\mathbf{K}_c}, \quad (9)$$

which includes two matrices \mathbf{K}_c and \mathbf{K}_g . \mathbf{K}_c denotes the so-called passive stiffness which simply describes the elastic behaviour of the cable and might also include the stiffness of the winch position controller. \mathbf{K}_g is the active stiffness of the system. Contrarily to the passive stiffness, it does not depend on elasticity effects such as compliance of a material. Instead, it reflects the attempts of the system to return to its force and torque equilibrium position once the platform is subject to a position disturbance. This behaviour only depends on the geometry and pose of the system, condensed in \mathbf{A}^T .

Now compliance matrix \mathbf{C} allows to compute the displacement for a given disturbance wrench.

$$\delta \mathbf{x}_P = \mathbf{K}^{-1} \delta \mathbf{w} = \mathbf{C} \delta \mathbf{w} \quad (10)$$

As this includes both translational and rotational motions, a diagonal homogenization matrix \mathbf{J}_v according to [15, 16] can be employed, i.e. $\mathbf{A}_h^T = \mathbf{J}_v^{-1} \mathbf{A}^T$, which can be used instead of \mathbf{A}^T in eq. (9), allowing to apply eq. (10) to get a homogenized displacement $\Delta \mathbf{x}_{Ph}$. The magnitude of this displacement $\kappa = \|\Delta \mathbf{x}_{Ph}\|_2$ gives a scalar measure for the stiffness which can be used as part of a cost function. Within this paper, the disturbance force \mathbf{w} includes two wrenches: First, the gravity is included as it leads to relevant position deviations, especially in upper parts of the workspace, see Figure 3 and Figure 4. Second, it includes the influence of constant cross wind of $7 \frac{m}{s}$ (4 Bft), modeled by the drag equation. Cross wind might lead to hardly predictable oscillations of the platform. For both wrenches, a high stiffness is desirable for the generated trajectories to minimize their effects.

Let $\kappa(q)$ be the displacement criterion at the q^{th} discretization step of one spline. The sum of $\kappa(q)$ over one spline segment is weighted using the time per segment.

$$\kappa_w(k) = \frac{t_k}{t_{\text{total}}} * \frac{1}{n_k} * \sum_{q=1}^{n_{sk}} \kappa(q) \quad (11)$$

Following, the total cost for stiffness throughout the whole trajectory is calculated based on the weighted sum $\kappa_w(k)$.

$$\mathcal{V}_{\text{stiff}} = \sum_{k=1}^{n_k} \kappa_w(k) * \frac{1}{n_s} \quad (12)$$

As there is no bound defined for stiffness in this work, $\mathcal{P}_{\text{stiff}} = 0$. In future work, requiring a minimum stiffness could be easily considered.

6.3 Collision Detection

As cable robots might use a high number of cables, collisions and their avoidance are crucial. The methods in this section were derived from [17, 18]. Within this work, only collisions between the lower cables and obstacles as well as between platform and objects are considered. In this simulation, the only obstacles on the site are the bricks which might be either on a pallet delivered by the supply chain, or already placed as part of the building. Accordingly, the poses of all objects on this simplified site are known. Generally, also collisions between the cables with each other and between cables and platform can be taken into account, but as the application at hand avoids platform rotations and uses a simple geometry – where the corners of the platform are connected to the corresponding

corners of the frame, avoiding interference – those cases are neglected.

The employed approach bases on the Separated Axis Theorem as presented by [18]. Here, obstacle bounding volume and end-effector bounding volume are calculated by axis-aligned bounding boxes (AABB).

Using the AABB approach, intersection tests in all axes are simple. Furthermore, the Euclidean distance d between two objects can be determined. Noteworthy, in this work, the distance is measured using two approaches: For measurements between end-effector and obstacles, face-to-face distances $d_{ee}(q)$ are computed. Face-to-center distances are applied for measurements $d_{ci}(q)$ between each i^{th} cable and obstacles. In case of multiple obstacles, the smallest distance to the nearest obstacle in each discretization step is taken into account for the cost function. If any of the distances falls below the minimum distance d_{min} or w_{min} (see table 3), respectively, costs are applied to avoid collisions within the trajectory. To calculate the cost in each discretization step q , the distances d are fed to an arbitrarily chosen nonlinear function

$$\mathcal{D}_{\text{EEOB}}(q) = \left(3 \left(1 - \frac{d_{ee}(q)}{d_{\text{min}}}\right)\right)^3 \quad (13)$$

which avoids the objects getting close to each other. The total costs for collisions between end effector and objects are given by the sum

$$\mathcal{V}_{\text{EEOB}} = \frac{1}{n_s} \sum_{k=1}^{n_k} \sum_{q=1}^{n_{sk}} \mathcal{D}_{\text{EEOB}}(q). \quad (14)$$

Similarly, the cost for collisions between cables and objects are calculated by an arbitrarily chosen nonlinear function that basically multiplies a weight to eq. (13)

$$\mathcal{D}_{\text{CAOB}}(q) = \sum_{i=1}^m \left(3 \left(1 - \frac{d_{ci}(q)}{w_{\text{min}}}\right)\right)^3 * \frac{1}{10} \quad (15)$$

and summarized as

$$\mathcal{V}_{\text{CAOB}} = \frac{1}{n_s} \sum_{k=1}^{n_k} \sum_{q=1}^{n_{sk}} \mathcal{D}_{\text{CAOB}}(q). \quad (16)$$

Whenever a step q along the path leads to a collision, penalties $\mathcal{P}_{\text{EEOB}}$ and $\mathcal{P}_{\text{CAOB}}$, respectively, are added each time. This holds for collisions between cables and objects as well as the end effector and objects. To save computation times, all bricks of a completed layer in the building are represented by a common box.

7 Discussion of Results

The models employed and the optimizer were set up using an installation plan for a small building. Table 3 lists

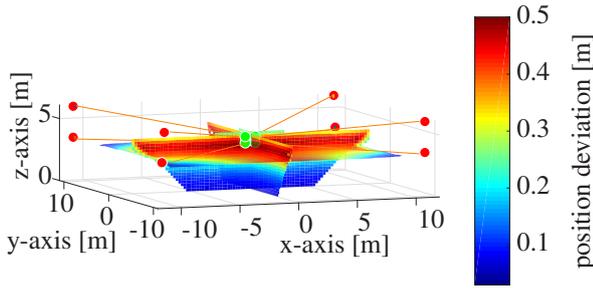


Figure 3. Stiffness in the workspace with lower pulleys upwards. The applied wrench $\delta\mathbf{w}$ includes crosswind of $7\frac{m}{s}$ (4 Bft) and platform weight of 100 kg, but no brick.

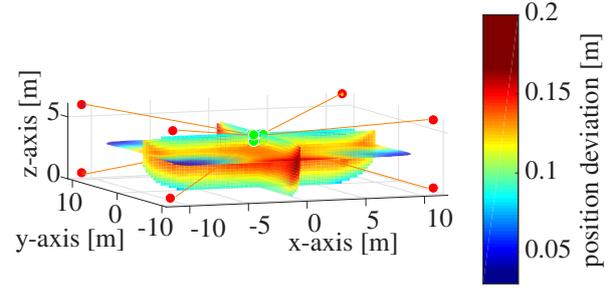


Figure 4. Stiffness in the workspace with lower pulleys downwards. The applied wrench $\delta\mathbf{w}$ includes crosswind of $7\frac{m}{s}$ (4 Bft) and platform weight of 100 kg, but no brick.

Table 2. Costs \mathcal{V} and weights \mathcal{W} . *a) without stiffness optimization. *b) with stiffness optimization. *c) save overhead trajectory

\mathcal{V}	*a	*b	*c	\mathcal{W}
$\mathcal{V}_{\text{time}}$	11.8732	11.2863	39.0000	0.5
$\mathcal{V}_{\text{force}}$	2.8247	3.0550	6.0302	1
$\mathcal{V}_{\text{EEOB}}$	0.0001	0.0021	0.0000	1
$\mathcal{V}_{\text{CAOB}}$	0.0000	0.0000	0.0000	1
$\mathcal{V}_{\text{pulleys}}$	0.0338	0.1100	0.0000	1
$\mathcal{V}_{\text{stiff}}$	(0.1234)	0.1205	(0.2237)	50
$\mathcal{V}_{\text{total}}$	8.7952	14.8369	25.5302	

$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$	$\begin{bmatrix} 0.5894 \\ 0.8683 \\ 2.4043 \end{bmatrix}$ s	$\begin{bmatrix} 0.4747 \\ 1.3457 \\ 1.8655 \end{bmatrix}$ s	$\begin{bmatrix} 3.0000 \\ 6.0000 \\ 3.0000 \end{bmatrix}$ s
---	--	--	--

the applied parameters. For computational efficiency during the optimization, only few time steps were used within the prime calculation of the optimizer, see table 3. Noteworthy, for higher accuracy, a final computation including a penalty check was carried out with a increased resolution of $[n_{s_1}, n_{s_2}, n_{s_3}] = [30, 29, 29]$. For the numerical optimization, MATLAB's particleswarm() with a hybrid setting – which basically appends a gradient decent using fmincon() – was chosen. These methods support parallel computing to speed up the simulation. The swarm population size was set to 60 and the maximum iterations to 1000.

Noteworthy, the moving masses of the platform change if a brick is picked. In optimization procedure, mass and inertia of both platform and brick are considered in the dynamics model, transformed with respect to \mathbf{x}_P located at the bottom of the brick. The results reflect the compromise found as a consequence of the – partly contrary – cost functions. Under the influence of platform and payload mass in gravity, the lower parts of the workspace are

favorable as they provide an advantageous angle of attack for the cables and thus lower cable forces.

A trajectory not considering stiffness is shown in Figure 5. Here, brick no. 300 was arbitrarily chosen, starting its motion at $\mathbf{r}_{\text{start}} = [-5.747, -0.876, 0.748]^T$ and placing the brick at $\mathbf{r}_{\text{goal}} = [0.0875, 6.922, 0.748]^T$ after a motion time of 3.8619 s.

As introduced in section 4, the total cost function chosen in this work is

$$\mathcal{V}_{\text{total}} = \sum_{\chi} \mathcal{W}_{\chi} \times \mathcal{V}_{\chi} + \mathcal{P}_{\chi} \quad (17)$$

where $\chi \in \{\text{time, traj, force, EEOB, CAOB, pulleys}\}$.

On the other hand, stiffness influences the result of the optimizer, both regarding platform trajectory and lower pulley motion. This is illustrated in Figure 3 and Figure 4. Obviously, the stiffness massively varies within the workspace, and the pulley position has a major impact. As a rule of thumb, the lower pulleys down configuration shown in Figure 4 has clear advantages over the lower pulleys up configuration shown in Figure 3. Noteworthy, the active stiffness term \mathbf{K}_g takes advantage from high cable forces which on one hand, contradicts $\mathcal{V}_{\text{force}}$, see eq. (7), and on the other hand increases stiffness at the workspace boundaries. Still, eq. (17) applies where now $\chi \in \{\text{time, traj, force, EEOB, CAOB, pulleys, stiff}\}$. Accordingly, including $\mathcal{V}_{\text{stiff}}$ in $\mathcal{V}_{\text{total}}$, the motion of platform and pulley for brick no. 300 changes, as shown in Figure 5. Table 2 shows the resulting cost functions in comparison. An interesting finding is, that for the chosen parameters settings, the trajectory optimized for stiffness as well is even faster requiring 3.6859 s, but at the price of generally higher forces and faster pulley motions. Collisions were totally avoided in both cases along all trajectory steps. All penalty terms were zero upon convergence of the optimizing process.

While the spline-based approach provides maximum flexibility for optimization, the resulting motions might be

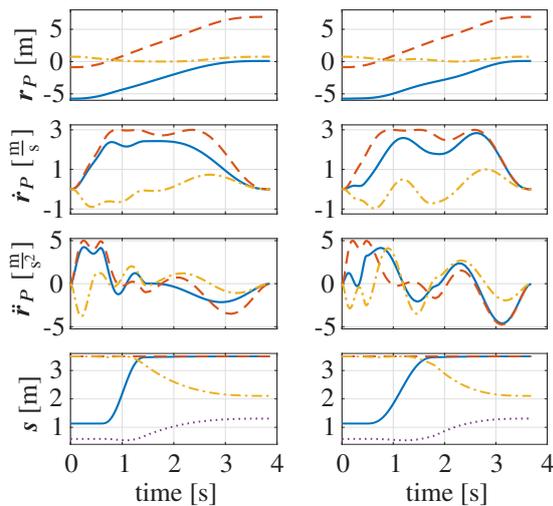


Figure 5. Optimized trajectories and pulley motion. Left: without stiffness optimization. Right: with stiffness optimization. Top three plots: Solid blue line: x -component of the vector. Dashed red line: y -component of the vector. Dash-dotted orange line: z -component of the vector. Bottom plots: Pulleys [1, 2, 5, 6]: Solid blue line, dashed red line, dash-dotted orange line, dotted magenta line.

unintuitive for any potential personnel on the site. Thus, in parallel, a slow simplified, unoptimized straight line trajectory over head level with fixed time was performed that is collision-free: It vertically lifts the brick to a height of 2 m above current wall height, horizontally moves to the goal position and vertically moves down to place the brick. The pulleys remain at the current wall height plus s_{\min} . For completeness, the results are given in table 2 as well. Clearly, since this trajectory is fully predefined at low dynamics, it is not optimal in comparison.

8 Conclusion and Outlook

This contribution introduced the models, the optimization problem and the solution approach to optimize the trajectories of a cable robot for automated bricking. The results indicate that the choice of the cost functions and their weights have a major impact on the trajectory and allow a tuning regarding preferences like transport time, stiffness and cable tension level. Collisions and paths outside the workspace could be effectively avoided.

This work will be extended by additional spline formulations and more effective parallelization in the future. The generated trajectories will be tested on real prototypes.

Acknowledgment

This research received funding by the Federal Ministry for Economic Affairs and Energy based on a resolution

of the German Bundestag. This funding is within the project “Entwicklung von Seilrobotern für die Erstellung von Kalksandstein-Mauerwerk auf der Baustelle” (IGF Vorhaben Nr.: 20061 BG), supported by the German Federation of Industrial Research Associations (AiF).

References

- [1] Jürgen Andres, Thomas Bock, Friedrich Gebhart, and Werner Steck. First results of the development of the masonry robot system rocco: A fault tolerant assembly tool. In *The 11th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 87–93. The International Association for Automation and Robotics in Construction (IAARC), 1994.
- [2] Günther Pritschow, Michael Dalacker, J. Kurz, and Jörg Zeiher. A mobile robot for on-site construction of masonry. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 3, pages 1701–1707. IEEE, 1994. ISBN 0-7803-1933-8. doi:10.1109/IROS.1994.407628.
- [3] Volker Helm, Selen Ercan, Fabio Gramazio, and Matthias Kohler. In-situ robotic construction: Extending the digital fabrication chain in architecture. In *Synthetic Digital Ecologies: Proceedings of the 32nd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, pages 169–176. CUMINCAD, 2012.
- [4] Tobias Bonwetsch. *Robotically assembled brickwork – Manipulating assembly processes of discrete elements*. PhD thesis, ETH Zürich, 2015.
- [5] Eric Brehm. Robots for masonry construction – status quo and thoughts for the german market. *Mauerwerk*, 23(2):87–94, 2019. doi:10.1002/dama.201900004.
- [6] Tobias Bruckmann, Hannah Mattern, Arnim Spengler, Christopher Reichert, Alexander Malkwitz, and Markus König. Automated construction of masonry buildings using cable-driven parallel robots. In *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction*, volume 33, pages 332–340. The International Association for Automation and Robotics in Construction (IAARC), 2016. doi:10.22260/ISARC2016/0041.
- [7] José Pedro Sousa, Cristina Palop, Eduardo Moreira, Andry Maykol Pinto, José Lima, Paulo Costa, Pedro Costa, Germano Veiga, and A. Paulo Moreira. *The SPIDERobot: A Cable-Robot System for*

Table 3. Values of the simulation parameters.

$\mathbf{P} = [{}^P\mathbf{p}_1 \dots {}^P\mathbf{p}_8] = \begin{bmatrix} 0.415 & -0.415 & 0.415 & -0.415 & 0.415 & -0.415 & 0.415 & -0.415 \\ -0.24 & -0.24 & -0.24 & -0.24 & 0.24 & 0.24 & 0.24 & 0.24 \\ 0.7 & 0.7 & 1.21 & 1.21 & 0.7 & 0.7 & 1.21 & 1.21 \end{bmatrix} \text{ m}$			
$\mathbf{B} = [{}^B\mathbf{b}_1 \dots {}^B\mathbf{b}_8] = \begin{bmatrix} 11.1 & -11.1 & 11.1 & -11.1 & 11.01 & -11.1 & 11.1 & -11.1 \\ -10.185 & -9.6375 & -10.185 & -10.185 & 9.6375 & 10.185 & 10.185 & 10.185 \\ 0 & 0 & 6.0 & 6.0 & 0 & 0 & 6.0 & 6.0 \end{bmatrix} \text{ m}$			
$\mathbf{J}_v = \text{diag}(1, 1, 1, 0.415, 0.24, 0.955)$	${}^B\mathbf{r}_S = [0 \ 0 \ 0.576]^T \text{ m}$	$m_P = 121.5264 \text{ kg}$	$\mu_f = 0.1$
$[n_{s_1}, n_{s_2}, n_{s_3}] = [10, 9, 9]$	$f_{\min} = 150 \text{ N}$	$f_{\max} = 9000 \text{ N}$	$d_{\min}, w_{\min} = 0.1 \text{ m}$
${}^B\boldsymbol{\theta}_P = \begin{bmatrix} 57.405 & 49.331 & 49.331 \\ 49.331 & 53.639 & 49.331 \\ 49.331 & 49.331 & 7.8261 \end{bmatrix} \text{ kgm}^2$	$[t_{\min}, t_{\max}] = [0.1, 15] \text{ s}$	$[s_{\min}, s_{\max}] = [0.5, 3.5] \text{ m}$	
$[{}^r\mathbf{r}_{P_{\min}}, {}^r\mathbf{r}_{P_{\max}}] = \begin{bmatrix} -11.100, 11.100 \\ -10.185, 10.185 \\ 0, 6 \end{bmatrix} \text{ m}$	$[{}^r\dot{\mathbf{r}}_{P_{\min}}, {}^r\dot{\mathbf{r}}_{P_{\max}}] = \begin{bmatrix} -3, 3 \\ -3, 3 \\ -3, 3 \end{bmatrix} \frac{\text{m}}{\text{s}}$	$[{}^r\ddot{\mathbf{r}}_{P_{\min}}, {}^r\ddot{\mathbf{r}}_{P_{\max}}] = \begin{bmatrix} -5, 5 \\ -5, 5 \\ -5, 5 \end{bmatrix} \frac{\text{m}}{\text{s}^2}$	

On-site Construction in Architecture, pages 230–239. Springer International Publishing, 02 2016. ISBN 978-3-319-26376-2. doi:10.1007/978-3-319-26378-6_17.

- [8] Andry Maykol Pinto, Eduardo Moreira, José Lima, José Pedro Sousa, and Pedro Costa. A cable-driven robot for architectural constructions: A visual-guided approach for motion control and path-planning. *Autonomous Robots*, 41(7):1487–1499, 10 2017. ISSN 1573-7527. doi:10.1007/s10514-016-9609-6.
- [9] Tobias Bruckmann, Lars Mikelsons, Thorsten Brandt, Manfred Hiller, and Dieter Schramm. Wire robots part I - kinematics, analysis & design. In Jee-Hwan Ryu, editor, *Parallel Manipulators - New Developments*, ARS Robotic Books, pages 109–132. IntechOpen Limited, London, 4 2008. doi:10.5772/5365.
- [10] Hubert Hahn. *Rigid Body Dynamics of Mechanisms: 1 Theoretical Basis*. Springer, 1 edition, 2002. doi:10.1007/978-3-662-04831-3.
- [11] Dieter Schramm, Manfred Hiller, and Roberto Bordini. *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen*. Springer Vieweg, 3 edition, 2018. ISBN 978-3-662-54481-5. doi:10.1007/978-3-662-54481-5.
- [12] Katharina Müller, Christopher Reichert, and Tobias Bruckmann. Analysis of geometrical force calculation algorithms for cable-driven parallel robots with a threefold redundancy. In Jadran Lenarčič and Oussama Khatib, editors, *Advances in Robot Kinematics*, pages 203–211, Cham, 2014. Springer International Publishing. ISBN 978-3-319-06698-1. doi:10.1007/978-3-319-06698-1_22.
- [13] Christopher Reichert, Paul Glogowski, and Tobias Bruckmann. Dynamische Rekonfiguration eines seilbasierten Manipulators zur Verbesserung der mechanischen Steifigkeit. In Torsten Bertram, Burkhard Corves, and Klaus Janschek, editors, *Fachtagung Mechatronik 2015: Dortmund (12.03.-13.03.2015), Tagungsband*, pages 91–96, Aachen, 2015. Institut für Getriebetechnik und Maschinendynamik. ISBN 978-3-00-048814-6. doi:10.17877/DE290R-7388.
- [14] Teun Hoevenaars. *Parallel manipulators with two end-effectors: Getting a grip on Jacobian-based stiffness analysis*. PhD thesis, Delft University of Technology, 2016.
- [15] Leo Stocco, S.E. Salcudean, and Farrokh Sassani. Matrix normalization for optimal robot design. In *1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 2, pages 1346–1351, 06 1998. ISBN 0-7803-4300-X. doi:10.1109/ROBOT.1998.677292.
- [16] Dinh Quan Nguyen and Marc Gouttefarde. Stiffness matrix of 6-dof cable-driven parallel robots and its homogenization. In Jadran Lenarčič and Oussama Khatib, editors, *Advances in Robot Kinematics*, pages 181–191, Cham, 2014. Springer International Publishing. ISBN 978-3-319-06698-1. doi:10.1007/978-3-319-06698-1_20.
- [17] Mario Lehmann. *Bahnplanung für Seilroboter unter Berücksichtigung von Kollisionen*. Bachelor thesis, University of Duisburg-Essen, 2019.
- [18] Christer Ericson. *Real-Time Collision Detection*. CRC Press, 2004. ISBN 978-1-558607323.