

Article

Simulation-Based Comparison of Novel Automated Construction Systems

Lukas Herrmann, Roland Boumann ^{*} , Mario Lehmann, Samuel Müller and Tobias Bruckmann 

Chair of Mechatronics, University of Duisburg-Essen, Forsthausweg 2, 47057 Duisburg, Germany

* Correspondence: roland.boumann@uni-due.de

Abstract: As automated construction processes require large workspaces and high payloads, the use of cables is a reasonable approach to span wide distances and share loads. Therefore, a cable-driven parallel robot is a suitable choice for automated masonry construction. Another possible robotic system for this task consists of a set of cooperative drones, each connected to the end effector and the payload by a cable. Because of the similarities between the two robotic systems, the same object-oriented programmed software can be used for trajectory planning and subsequent investigations, making minor adjustments. The implemented optimizing path planning algorithm takes into account the physical boundaries, motion time, collision avoidance and energy requirements. Thus, a simulation-based comparison of the characteristics of both systems can be made. In this paper, the necessary physical models for both the drone system and the cable robot are derived in detail. Based on the common framework, this paper presents a comparison between the two robotic systems, defining two different scenarios. The first scenario demonstrates the functioning of the optimizer approach. The second scenario is used to compare the two systems. For this purpose, the trajectories for all 1720 masonry units of the first floor of a house are optimized. The analysis of the results shows that both systems can transport heavy loads, with the cable robot having advantages on smaller sites, while the drone system covers larger distances for the price of slower performance and higher energy consumption.

Keywords: cable robot; drone cooperation; numerical optimization; path planning; robotic construction; construction automation; automated masonry; modeling and simulation



Citation: Herrmann, L.; Boumann, R.; Lehmann, M.; Müller, S.; Bruckmann, T. Simulation-Based Comparison of Novel Automated Construction Systems. *Robotics* **2022**, *11*, 119. <https://doi.org/10.3390/robotics11060119>

Academic Editor: Thomas Bock

Received: 3 August 2022

Accepted: 21 October 2022

Published: 31 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction and State of the Art

In the modern component manufacturing industry, the use of CAx software is widespread, especially in automated series production. The advantages are an improved quality, reduced costs and shorter manufacturing time. In contrast, the design and construction of buildings are mostly individual and thus still require the execution of work on site, typically based on traditional manual processes using printed two-dimensional technical drawings. However, in recent years, digital process chains are also being developed in the construction of modern buildings using the Building Information Modeling (BIM) method [1]. Analogous to the process in component manufacturing, one core method of BIM is the digital modeling of buildings. The modeling in BIM is based on parameterizable objects, functions and logics. Further advantages for the user include the creation of requirements and drawing plans, the possibility to calculate costs, simplified scheduling and the visualization of the building. At the end of the modeling process, manufacturing data can be generated from the digital models. These data sets might serve in the future as the basis for automating certain work processes and executing them by using different robots [2,3].

Notably, a major challenge in the design of automated construction systems is the extremely large volume of typical sites, which has to be covered by the automated construction solutions. Conventional robot arms fail, which is why special robot arms of

large scale or mobile platform bases have been presented [4]. Furthermore, the typical mass of construction elements is a challenge for those robot systems employing serial kinematic structures.

Alternatively, the use of multiple cables is an efficient approach to share payloads and to span large distances. Therefore, the concept of automated masonry using a cable-driven parallel robot (CDPR) has been presented recently [5,6]; see Figure 1a. CDPRs are a special form of a parallel manipulator and thus share fundamental kinematical properties with hexapods. The workspace of the CDPR is spanned by a mobile frame which makes it attractive for construction processes. Within this workspace, the end effector of the robot can be freely positioned in all six degrees of freedom by individually changing the length and tension of the cables. The cables run from motor-controlled cable winches via guiding pulleys to anchor points on the end effector. The guiding pulleys are attached to the frame, either fixed or on movable mechanisms. The actual task of the cable robot for masonry works is then to pick up the bricks e.g., at an automated mortar system, and to place them in the defined poses according to BIM model data. Typically, this should be done in the shortest possible time. The CDPR has two properties that make it particularly suitable for this application. On the one hand, overactuation (e.g., using more actuated cables than the end effector's number of Degrees-of-Freedom) can be used to enhance the stiffness of the system, which is crucial for outdoor applications. On the other hand, the movable masses are extremely low as these are essentially limited to the end-effector and the payload. Still, a CDPR has drawbacks, where the most obvious drawback is the high number of cables that may even run below the end effector level, which introduces a risk of collisions with already built structures or moving objects. This has to be handled, e.g., by vertically moving pulleys. Moreover, cables are unilateral constraints in general. This needs to be considered while path planning and control. In addition, the purchasing costs as well as the setup costs for the assembly and disassembly of the cable robot must be taken into account, when checking the economic feasibility [7]. Still, a first prototype for automated masonry has been presented and technically validated [8,9] as shown in Figure 1a, including an automated mortar station which delivers masonry units to the CDPR using a lifter. Notably, this prototype is designed for single-storey experiments and thus only the lower pulleys are actuated to move vertically. Due to their advantages, CDPR have been investigated also for other relevant construction processes such as 3D printing of concrete [10–14] or curtain wall installation as demonstrated in the EU project, Hephaestus [15].

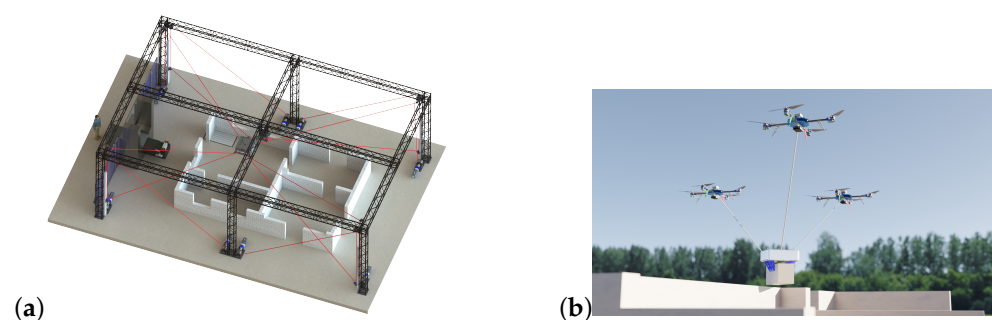


Figure 1. A cable-driven robot in (a) and a drone system in (b) used for the autonomous masonry of a building.

In contrast, a multicopter or drone is a spatially unlimited applicable robotic system with comparatively low set-up costs. Since the beginning of their development, multicopters have been used to transport lightweight objects. A significant application example is a camera-carrying drone. In the film and advertising industry, drones are now predominantly used instead of helicopters because they offer much more versatile camera settings and are significantly less expensive. At large events, camera drones are used by the police and event organizers to monitor the event area and direct the stream of visitors. Another application scenario, although not yet as widespread, can be found in agriculture. Here, drones are used for example as an equipment carrier for pest control [16]. In the

construction industry, drones are primarily used for land surveying, documenting the progress of construction work or assessing damage of hard-to-reach places or areas [17–19]. Additionally, they are used for active monitoring of construction sites to achieve an increase in construction site safety. For an overview, see [20]. All previously mentioned use cases have in common that the payload of the drone is comparatively small.

Over the years, however, the payload for drones increased continuously, so that, nowadays, heavy-lift drones with a maximum payload of up to 100 kg can be purchased. The reason for this are, among other things, increasingly better lithium-ion batteries, which enable the use of more powerful motors [21]. Due to these developments, drones are thus becoming increasingly interesting for the transportation of heavy loads or for the use in internal logistics [22]. Research in the field of alternative energy sources, such as the use of a fuel cell, may increase the operating time in the future [23]. A multicopter might be equipped with an end effector that is able to pick up a payload. Essentially, three different approaches are pursued; see [24]. In the simplest case, the end effector is firmly connected to the drone [25–28]. However, when using underactuated drones, this method has the disadvantage that the payload cannot be fully oriented. One way to overcome the underactuation is to use a rigidly mounted end effector in combination with a full- or overactuated multicopter [29]. A different approach discussed in [30,31] is the use of drones with robotics arms. This second approach also allows the independent movement of the payload in all six degrees-of-freedom. The approaches described so far are particularly suitable for the transport and/or manipulation of lightweight objects.

One disadvantage, however, is that the payload is usually located very close to the drone. This might be an issue in specific applications (e.g., in cluttered environments). Therefore, another frequently investigated use case is the transport of a payload that is connected to a drone via a cable. This allows the adjustment of the distance between the drone and the payload almost arbitrarily. As a major advantage, by using multiple, cooperating drones connected to a common payload using cables, the maximum payload weight can be drastically increased and the maneuverability can be enhanced. In the past, drone cooperation has already been investigated repeatedly. In [32], the transport of a payload, which is still considered as a point mass, is investigated using two drones. However, this does not increase the controllability and, furthermore, the drones have to be far enough apart. If the distance of the drones is increased, the cable forces increase as well since the system is closer to a singularity of the second kind. The transport of a rigid beam using two quadcopters is demonstrated in [33]. With the use of three or more drones, the payload is kinematically underconstrained [34] but statically determined with the help of gravity. Moreover, it can be fully controlled, as the three drones in total have more controllable degrees of freedom than the end effector. As examples, flight maneuvers with three drones are presented in [35–37]. When using three or more drones, there is also the possibility to reconfigure them in space without moving the payload. With these maneuvers in null space, the properties of the system can be influenced. This fact and the relation to cable robots is considered in [38]. The goal of planning an optimal trajectory in terms of the cable angles and energy efficiency is considered in [39]. Path planning under external disturbances is shown in [40]. Planning a trajectory for the specific task of flying through a window is the subject of the work [41]. In addition, [42] deals with highly dynamic trajectories. The procedure of lifting the payload is analyzed in [43,44].

In the scope of this paper, another possible scenario would be using cooperating drones on large construction sites to handle internal logistics or to do simple construction tasks, such as the automated masonry already mentioned, see Figure 1b. Unlike cable robots, drones are not locally bound. Furthermore, the space and infrastructure requirements on the construction site are easy to provide.

Accordingly, while both systems use cables to share the load of high payloads, they still have inherent technological differences, leading to differing characteristics when used for automated masonry. As for this application, both systems have only been tested in very early technological stages, but might be very promising to be developed within

future research projects, it is reasonable to create approaches to evaluate and compare their potential performances in automated masonry using simulation tools. Thus, this contribution aims at comparing the two presented cable-based systems in predefined scenarios using numerical model-based simulations. Due to the model-based approach, the physical models for both the drone system and the cable robot need to be derived in detail. Based on a software framework, the two robotic systems are evaluated, defining two different scenarios. The first scenario demonstrates the applicability and capabilities of the optimizer approach. The second scenario is used to compare the two systems in terms of several performance indices such as required construction time and energy requirements.

The paper is structured as follows: In Section 2, the methodical approach and the used optimization framework are presented. Section 3 introduces the cost functions used for the path planning algorithm. The approach is then demonstrated in Section 4 to optimize trajectories for a cable robot and a system of cooperating drones. The optimization results and the different characteristics of both systems are discussed. The paper is summarized in Section 5.

2. Methodological Approach

2.1. Setup of the Software Framework

As introduced, the analysis of the robot systems is supported by a software framework implemented as a generic trajectory optimization tool. To be as flexible as possible, the framework uses an object-oriented architecture. Through the object-based software design the workflow of the transport of a masonry unit can be split up into several trajectories, in which each movement such as lifting, moving to a target position, placing down and moving back to the starting position becomes an own independent optimization problem with its own parameterization. Note that BIM data are used to provide the information required for this purpose, which among other things, include the target position and orientation of each brick used in the building. For this purpose, a BIM postprocessing algorithm was created to derive masonry laying plans from a given BIM model.

Figure 2 shows the general workflow using the framework.

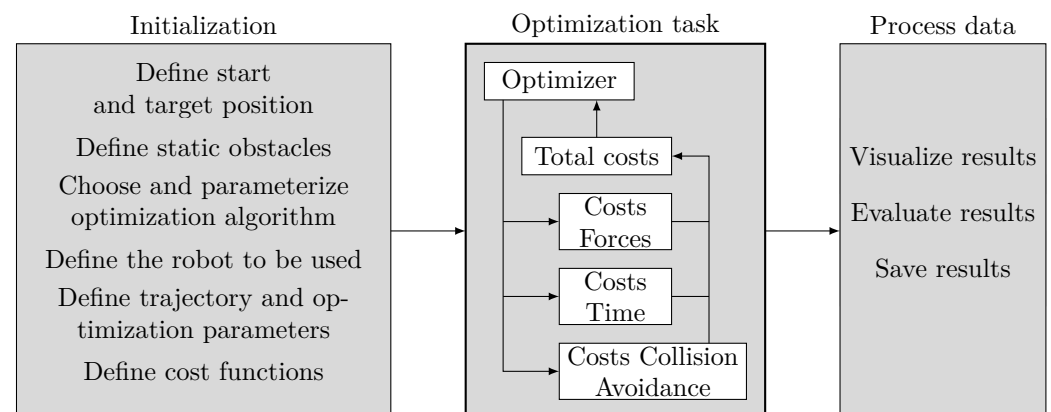


Figure 2. Key steps in optimization process.

The core of the framework includes a number of optimization algorithms, both from the Optimization Toolbox and the Global Optimization Toolbox of MATLAB® [45]. For trajectory optimizations, which require the tuning of multiple parameters per spline-segment, a global optimizer is useful, such as a Particle Swarm Optimizer. On the other hand, a gradient-based optimizer might reduce computational costs and can be applied in special cases, e.g., when a spatial fixed trajectory should be time-optimized only. Based on this, cost functions representing technical criteria can be created. These can be evaluated for a number of trajectories. The properties and parameters of each trajectory can be defined independently, including the definition of the start and target position (e.g., based on the BIM data provided), the specification of the robot used and its properties, the optimized

parameters, the cost functions to be applied per trajectory, the static objects that must be considered for collision avoidance as well as the definition of the payload. For each trajectory, the parameters and properties are updated before the optimization starts. As multiple cost functions might be needed, they are added to a single overall cost function using weights which allows to vary their influence.

Due to the object-oriented approach, only the models for the individual physical behavior of the robot and the corresponding cost functions have to be adapted to compare automated construction system for a given masonry laying plan.

2.2. Parametrization of Trajectories

A trajectory describes the motion of the robot and its components over time. Accordingly, it contains information about the pose and its time derivatives. In general, the trajectory will be optimized in the world reference frame based on the framework in Section 2.1. The mathematical description of the trajectory must be defined in a consistent manner that allows the variation of its parameters by the chosen numerical optimizer. For this issue, splines—i.e., a chain of interconnected polynomials with common knots—are a popular choice. The coefficients of the polynomials are determined by the numerical solution of a system of equations $Ax = b$, whereby the vector b contains initial variables, some of which are optimization parameters. By changing these parameters using an optimization algorithm, a different trajectory is generated which can be evaluated with respect to the cost functions. The total costs are the sum of the individual costs, where each individual cost function evaluates a single technical criterion of the trajectory such as time, energy requirements and avoidance of collisions.

To allow for a precise masonry process including the required intermediate pose measurements, alignment processes and masonry process steps (such as application of mortar), the transportation task is decomposed into 15 consecutive segments for each masonry unit. For each partial step, an individual trajectory is created and registered for the optimization process. Table 1 gives an detailed overview.

Table 1. Transportation sequence for each masonry unit.

Trajectory	Function	Payload	Time [s]
1	Measuring the dimensions of the masonry unit	no	32.7
2	Slowly move down to the pickup position	no	9.12
3	Pick up the masonry unit	no	5.54
4	Slowly move up half a meter	yes	4.89
5	Move up to transition height	yes	optimized
6	Transition through workspace to target	yes	optimized
7	Move down to half a meter above target position	yes	optimized
8	Measuring the target position	yes	20.64
9	Slowly move down half a meter	yes	2.62
10	Placing down the masonry unit	yes	11.78
11	Slowly move up half a meter	no	1.31
12	Move up to transition height	no	optimized
13	Transition through workspace	no	optimized
14	Move down to half a meter above next masonry unit	no	optimized
15	Wait until the mortar system has finished	no	externally defined

Essentially, the trajectories differ in whether the end effector is currently transporting a masonry unit or not and whether the time for a path segment is optimized or not. A

full collision check is performed for all path segments. In the case of the cable robot, both the pulleys of the lower cables and the mortar system must be raised with increasing height in order to avoid a collision between cables and the already completed part of the masonry which is also reflected by a trajectory computed for each moving pulley. In the measuring process during trajectory 1 and 8, the end effector is aligned over the start and target position using laser sensors. While most of the other trajectories are predefined due to process conditions, the duration of trajectory 15 is defined by the external mortaring system. In addition to a fixed time t_M , which the mortar system needs to apply the mortar, the lifting of the mortar system also requires a certain time t_H . The time $t_{c,15}$ that the cable robot has to wait is therefore calculated by

$$t_{c,15} = t_M + t_H - \sum_{i=1}^{14} t_i . \tag{1}$$

In case of the drone system, the risk of a cable collision with the already completed part of the masonry is reduced as there are no cables below the current masonry layer. Accordingly, the mortar system must not be lifted. Thus, the duration $t_{d,15}$ of the 15th trajectory is calculated for the drone system as

$$t_{d,15} = t_M - \sum_{i=1}^{14} t_i . \tag{2}$$

In case $t_{c,15}$ or $t_{d,15}$ becomes negative, the 15th trajectory is skipped and its duration is set to zero. As long as a single trajectory is independent from the other ones, the evaluation sequence is arbitrary which can be very beneficial for parallelization of the computations.

Dependent on the robotic system, up to $f_P = 6$ degrees-of-freedom (DOF) of the platform pose \hat{r}_P can be independently specified over time by splines. All trajectories with the exception of trajectories 6 and 13 have no additional knots between their start and target position and are therefore represented by a single segment. As for trajectories 6 and 13, a certain variability of the trajectory is demanded, but a high number of spline segments would increase the number of optimization variables. Hence, three spline segments were chosen to compromise between flexibility and computation time. For the trajectories, the following constraints are given:

- **At start:** Position is given, the velocity and all higher order time derivatives are set to zero, i.e., no optimization parameters.
- **At first and second knot:** Position, velocity and acceleration are parameters, subject to numerical optimization, i.e., three parameters in total, per DOF and knot. In the case of six DOFs, this leads to 18 parameters per knot. Jerk and higher order time derivatives are set to zero.
- **At goal:** Position is given, the velocity and all higher order time derivatives are set to zero, i.e., no optimization parameters.

A trajectory with n_k additional knots has $3f_P n_k$ optimization parameters for the description of the movement. As already introduced, the transition time per path segment can be defined as an optimization parameter. This adds up to $n_k + 1$ additional parameters. To evaluate the cost functions and do a proper collision check for a trajectory, each spline is discretized in n_s time steps along the spline where $t_s(k)$, $k \in \{1, \dots, n_s\}$ is a specific time step.

2.3. Definition of the Optimization Parameters

The resulting optimized trajectory for a robotic system massively depends on the specific technical criteria such as low transition time, low energy consumption and collision avoidance. Each property that should be taken into account or optimized is considered as a criterion χ , valued by costs at the end of each optimization step. The costs of each criterion χ are calculated as the sum of a cost value \mathcal{V}_χ , multiplied by the weight \mathcal{W}_χ , and a penalty

term \mathcal{P}_χ . While the cost value \mathcal{V}_χ rates a continuous optimization parameter, the penalty term \mathcal{P}_χ is only non-zero when physical boundaries are not met. Note that tasks (such as of collision avoidance) may include both summands. Furthermore, the weights contain the respective physical units, such that the costs for each χ become dimensionless. The value of the total costs $\mathcal{V}_{\text{total}}$ for a trajectory is computed by summing up each single costs of a criterion χ . Therefore, the task to find the best trajectory in terms of the defined cost functions can be formulated as the unconstrained optimization problem

$$\min \sum_{\chi} (\mathcal{W}_\chi \cdot \mathcal{V}_\chi + \mathcal{P}_\chi). \quad (3)$$

This approach allows the multi-criteria optimization procedure to have very different weightings in order to meet a wide range of priorities, such as productivity (time optimization), economic efficiency (energy optimization) or safety (collision avoidance). Moreover, it is simple to understand and enables for flexible tuning to represent individual preferences. Note that the criteria (cost functions) in general are within a conflict, since the improvement of one cost function at some point (e.g., a short transition time) will degrade another (e.g., cable forces). However, the criteria are physically independent and cannot be transformed into each other. Hence, the weighting factors are arbitrary and influenced by the subjectivity of the user. Moreover, the subjectivity influences the solution which is considered as optimal. A useful approach for future work would be the investigation of Pareto optimal methods. Here, a quantity of solutions to the optimization problem is determined, where an improvement of one cost value results in a degradation of another. This quantity, which consists of all optimal compromises, is called Pareto front. This front allows an individual selection of an optimal solution under given preferences for the optimization criteria [46].

Here, the following criteria are considered as cost functions:

- **Trajectory time:** The robot motion should be as fast as possible. This can be achieved by defining $\mathcal{V}_{\text{time}} = \sum t_s$. Since the time has no upper boundary, it follows $\mathcal{P}_{\text{time}} = 0$.
- **Maximum end effector position, velocity and acceleration:** For each time step in which the boundaries of the position, velocity or acceleration of the end effector are exceeded, the associated penalty term \mathcal{P}_{pos} , \mathcal{P}_{vel} or \mathcal{P}_{acc} is charged. The boundaries of the position are derived from the robot's frame, the ones of the velocity and acceleration depend on the motor specifications but can also be limited for safety reasons. A limitation in task space is intuitive.
- **Actuator forces:** In the case of the cable robot, the costs $\mathcal{V}_{\text{Force}}$ include the cable forces as a major criterion. The physical boundaries f_{min} and f_{max} of the cable forces should not be exceeded to avoid cable breaks and cable slackness, respectively [47]. Therefore, the penalty term $\mathcal{P}_{\text{force}}$ is added whenever a cable force violates the boundaries. In the case of the cable robot, the necessary cable forces are generated by the torque of the winch motors. As the winches have masses, they introduce their own dynamics and boundaries. Hence, the additional cost value $\mathcal{V}_{\text{torque}}$ as well as the penalty term $\mathcal{P}_{\text{torque}}$ are defined. Note that, for simplicity, the motor torque limits are considered constant. The modeling of complex motor effects, such as field weakening, iron losses or temperature dependencies is part of future work. In the case of the drone system, the forces are generated by propellers. This leads to the cost function \mathcal{V}_ω and the penalty term \mathcal{P}_ω .
- **Power and energy consumption:** An electrical power source has technically defined limitations. To protect the source from overload, the total power demand needs to be limited. This is taken into account by the penalty term $\mathcal{P}_{\text{power}}$. A cost value $\mathcal{V}_{\text{power}}$ might be used, e.g., to consider thermal load. Within this work such effects are neglected, i.e., $\mathcal{V}_{\text{power}} = 0$. The energy consumption is one out of many factors which influence the economic efficiency of a robotic system. Hence, this parameter can also be considered by a cost function $\mathcal{V}_{\text{energy}}$. Since there is often no physical boundary, it usually follows $\mathcal{P}_{\text{energy}} = 0$.

- **Self-collision of the robots:** For both robotic systems, the self-collision of cables are investigated. Whenever a cable collision is detected, the penalty term \mathcal{P}_{CaCa} is assigned to the costs. In the case of the drone system, the self-collisions of drones are considered as well and, if necessary, are taken into account by the penalty term \mathcal{P}_{DD} .
- **Collisions of the robot with its environment:** This includes collisions between objects in the workspace and
 - the end effector or payload (ObEE),
 - the drones (ObD) and
 - the cables (CaOb).

Note that collisions between cables and the end effector (CaEE) can be treated as CaOb collisions.

During the optimization process the penalty terms are used as follows: If a boundary violation for the considered cost function occurs, a constant penalty value is applied as given e.g., in Table 3. Consequently, the optimizer tends to move in the direction of penalty-free trajectories, i.e., solutions where $\mathcal{P}_\chi = 0$. At the end of the optimization, a penalty-free trajectory is considered valid.

3. Modeling and Cost Functions

This section discusses the fundamentals to represent the relevant technical criteria by appropriate cost functions for optimization of the trajectories. Both the cable robot and the multicopter drone system are presented.

3.1. Modeling of the Platform Dynamics

The feasibility of a trajectory depends on the ability of the robotic system to apply the required forces to the platform. The derivation of the necessary equations is based on the schematic illustration in Figure 3.

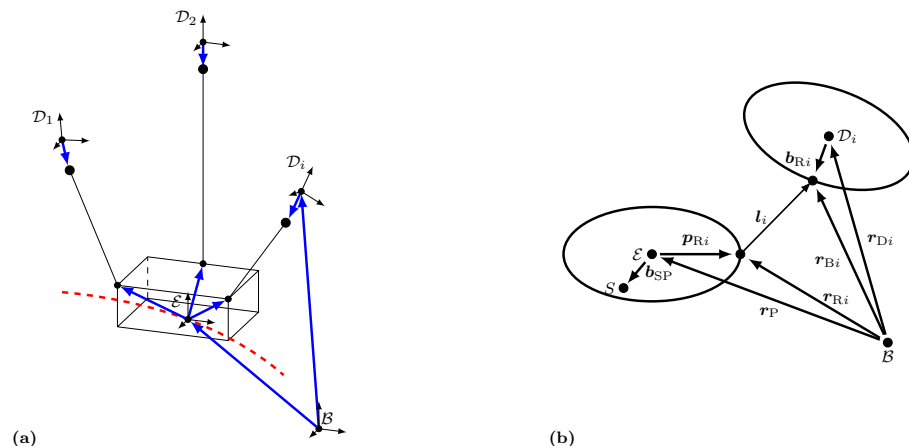


Figure 3. Schematic illustration of the transportation of a platform using cables in (a) and the definition of the necessary vectors in (b).

The unit consisting of the actual end effector—which is an actuated gripping device plus the required sensors (as shown e.g., in [8])—and the masonry unit to be transported is called a platform. Accordingly, its properties differ depending whether the platform is moving a masonry unit to its goal position or returning empty to the mortaring station, which is why they must be recalculated at the beginning of each trajectory. Each rigid body \mathcal{K} can be described by the Newton–Euler equation

$$\underbrace{\begin{bmatrix} m_{\mathcal{K}} E & \tilde{b}_{S\mathcal{K}}^T \\ m_{\mathcal{K}} \tilde{b}_{S\mathcal{K}} & \Theta_{\mathcal{K}} \end{bmatrix}}_{\tilde{M}_{\mathcal{K}}} \underbrace{\begin{bmatrix} \ddot{r}_{\mathcal{K}} \\ \dot{\omega}_{\mathcal{K}} \end{bmatrix}}_{\ddot{i}_{\mathcal{K}}} = \underbrace{\begin{bmatrix} -m_{\mathcal{K}} \tilde{\omega}_{\mathcal{K}} \tilde{\omega}_{\mathcal{K}} \tilde{b}_{S\mathcal{K}} \\ -\tilde{\omega}_{\mathcal{K}} \Theta_{\mathcal{K}} \omega_{\mathcal{K}} \end{bmatrix}}_{\hat{f}_{\mathcal{K}}} + \underbrace{\begin{bmatrix} f_{\mathcal{K}}^e \\ \tau_{\mathcal{K}}^e \end{bmatrix}}_{\hat{f}_{\mathcal{K}}^e} + \underbrace{\begin{bmatrix} f_{\mathcal{K}}^z \\ \tau_{\mathcal{K}}^z \end{bmatrix}}_{\hat{f}_{\mathcal{K}}^z}. \quad (4)$$

Now, the platform P is considered, i.e., $\mathcal{K} = P$. The platform dynamics with its mass m_P are described in the platform-fixed coordinate system \mathcal{E} . The origin of the coordinate system is centrally fixed below the gripping unit. Hence, if a masonry unit is fixed in the gripper, \mathcal{E} coincides with the center of the masonry unit’s bottom. Note that this holds for all masonry units within the building employed in this work. The inertia of the platform Θ_P with respect to the coordinate system \mathcal{E} is obtained considering the inertia of the end effector Θ_S and the masonry unit with respect to their centers of gravity and the Steiner–Huygens relation, if a masonry unit is gripped. The vector \mathbf{b}_{SP} points from the platform’s coordinate system to its center of mass. The matrix $\mathbf{E} \in \mathbb{R}^{3 \times 3}$ is an identity matrix. The second time derivative $\ddot{\hat{\mathbf{r}}}_P$ of the pose $\hat{\mathbf{r}}_P$ consists of the second time derivative of the position vector \mathbf{r}_P and the first time derivative of the angular velocity $\boldsymbol{\omega}_P$. The platform dynamics in Equation (4) can be summarized using the mass matrix $\hat{\mathbf{M}}_P$, the vector $\hat{\mathbf{f}}_P^c$, which is composed of the Coriolis and centrifugal force $\mathbf{f}_P^c = -m_P \tilde{\boldsymbol{\omega}}_P \tilde{\boldsymbol{\omega}}_P \mathbf{b}_{SP}$ and torque $\boldsymbol{\tau}_P^c = -\tilde{\boldsymbol{\omega}}_P \Theta_P \boldsymbol{\omega}_P$, the vector $\hat{\mathbf{f}}_P^e$, which is composed of the applied force \mathbf{f}_P^e and torque $\boldsymbol{\tau}_P^e$, and the vector $\hat{\mathbf{f}}_P^z$, which is composed of the constraint force \mathbf{f}_P^z and torque $\boldsymbol{\tau}_P^z$. Products of type $\tilde{\mathbf{a}}\mathbf{b}$ with the skew-symmetric matrix

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{5}$$

correspond to the cross product $\mathbf{a} \times \mathbf{b}$. With respect to Equation (4), the constraint force at the platform can be calculated for a given trajectory by

$$\hat{\mathbf{f}}_P^z = \hat{\mathbf{M}}_P(\hat{\mathbf{r}}_P)\ddot{\hat{\mathbf{r}}}_P - \hat{\mathbf{f}}_P^c(\hat{\mathbf{r}}_P, \dot{\hat{\mathbf{r}}}_P) - \hat{\mathbf{f}}_P^e(\hat{\mathbf{r}}_P, \dot{\hat{\mathbf{r}}}_P). \tag{6}$$

The constraint force $\hat{\mathbf{f}}_P^z$ must be applied by the cables, which are attached to the platform. The kinematic modeling (both for a CDP and the drone system) is similar to conventional Stewart Gough platforms. As rotation requirements of the platform are limited, simply roll-pitch-yaw angles have been applied. Kinematic modeling details can be found in [48]. Note that, for the drone system, the base points are moving. Each cable in the parallel mechanism is described by the vector $\mathbf{l}_i = \mathbf{r}_{B_i} - \mathbf{r}_P - \mathbf{p}_{R_i}$. Here, the vector \mathbf{p}_{R_i} points from the platform’s coordinate system to the attachment point R_i of the i -th cable, while the vector \mathbf{r}_{B_i} points from the origin of the world reference frame \mathcal{B} to the point B_i . The cable attachment point R_i can also be described with regard to the origin of the world reference frame \mathcal{B} by the vector \mathbf{r}_{R_i} . In the case of a cable robot, at point B_i the i -th cable enters the workspace. In the case of a drone system, the i -th cable is fixed at the i -th drone at this point. While in the case of the drone system the platform is connected to each drone by a cable of the constant length $l_{0,i} = \|\mathbf{l}_i\|_2$, the cable length is variable in the case of a cable robot. The resulting constraint conditions are given by the algebraic equation

$$\mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{n_C} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \left(\mathbf{l}_1^T \mathbf{l}_1 - l_{0,1}^2 \right) \\ \frac{1}{2} \left(\mathbf{l}_2^T \mathbf{l}_2 - l_{0,2}^2 \right) \\ \vdots \\ \frac{1}{2} \left(\mathbf{l}_{n_C}^T \mathbf{l}_{n_C} - l_{0,n_C}^2 \right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{7}$$

Equation (7) states that each of the n_C cables reduces the platform’s degrees-of-freedom by one. It is well known that the constraint forces can be expressed by $\hat{\mathbf{f}}^z = \mathbf{C}^T \boldsymbol{\lambda}$, where the matrix is calculated by $\mathbf{C} = \frac{\partial \mathbf{g}}{\partial \hat{\mathbf{r}}}$. Note that the constraints introduced by the cables are holonomic, since they are not velocity dependent. The result shows that the reaction forces point in the direction of the cables. Thus, each cable force \mathbf{f}_{R_i} can be described by the unit

cable vector \mathbf{v}_{Ri} multiplied by the magnitude of the cable force f_{Ri} . Therefore, the constraint forces can be expressed as

$$\hat{\mathbf{f}}_P^Z = - \underbrace{\begin{bmatrix} \mathbf{v}_{R1} & \cdots & \mathbf{v}_{Rn_C} \\ \tilde{\mathbf{p}}_{R1} \mathbf{v}_{R1} & \cdots & \tilde{\mathbf{p}}_{Rn_C} \mathbf{v}_{Rn_C} \end{bmatrix}}_{A^T} \begin{bmatrix} f_{R1} \\ \vdots \\ f_{Rn_C} \end{bmatrix}, \quad \text{with } \mathbf{v}_{Ri} = \frac{f_{Ri}}{\|\mathbf{f}_{Ri}\|_2} \tag{8}$$

$$\iff \hat{\mathbf{f}}_P^Z = - \underbrace{\begin{bmatrix} E & \cdots & E \\ \tilde{\mathbf{p}}_{R1} & \cdots & \tilde{\mathbf{p}}_{Rn_C} \end{bmatrix}}_{P^T} \begin{bmatrix} f_{R1} \\ \vdots \\ f_{Rn_C} \end{bmatrix}.$$

Since the assumed ideal cables cannot apply any torque, there are not any direct constraint torques. Equation (8) can be solved for the actual cable forces e.g., by using the Moore–Penrose pseudoinverse of P^T . Note that, for the cable robot, an active-set method is used to compute the scalar cable forces using A^T , which is a common approach for CDPRs [49]. As long as the end effector is in the workspace, the algorithm generates forces between the given limits f_{\min} and f_{\max} [49]. Further, note that the cable force calculation is a separate algorithm (e.g., based on active set optimization), which is called by the trajectory optimizer to validate a trajectory for feasible cable forces at each time step. Typically, when low forces are demanded, this will lead to cable forces close to or even at f_{\min} for some cables. Therefore, in practice, f_{\min} is chosen slightly above the physically required value to leave a margin for control purposes. Since all cables need a minimum tension f_{\min} to avoid slackness and obtain a valid solution, the costs for the cables forces can be taken into account by

$$\mathcal{V}_{\text{force}} = \frac{1}{n_C n_s} \left(\sum_{k=1}^{n_s} \sum_{i=1}^{n_C} \frac{f_{Ri}(k) - f_{\min}}{f_{\max} - f_{\min}} \right). \tag{9}$$

Every time the cable forces are beyond f_{\min} or f_{\max} a penalty term $\mathcal{P}_{\text{force}}$ is added and $\mathcal{V}_{\text{force}}$ is not assigned. Therefore, the costs for this criterion cannot be negative. Note that this also holds for similar cost functions. Considering singularities, three kinds need to be taken into account for parallel robots as explained in [49]. Serial singularities (first kind) can be neglected from a practical point of view, since the cable lengths are fixed for the drone system and will never drop to zero for the CDPR. Parallel singularities (second kind), which coincide with a loss of rank within the platform’s Jacobian A^T , will cause an error when trying to compute cable forces, leading to a penalty. Consequently, the optimizer will avoid such configurations throughout a trajectory, even if they exist. Moreover, the CDPR design has been analyzed for feasible cable forces throughout the required workspace [48]. Architectural singularities (third kind) are avoided by carefully choosing the design parameters of the system. For the drone, for example, the cables are arranged on the platform in such a way that two cables never converge at one point. Furthermore, the only desired rotation is about the vertical axis in order to place a masonry unit. Note that this is done by a rotational gripper in the case of the CDPR. This reduces the set of required configurations drastically from a practical point of view.

3.2. Modeling of the Drone Dynamics

The considered drone system consists of a platform $\mathcal{K} = P$ with the mass m_P and $i = 1, \dots, n_C$ drones D_i with the mass m_D and the inertia Θ_D . Within this work, we considered quadrotors for a first implementation and verification of the optimization algorithm. Based on the object-oriented structure of the optimization framework, the quadrotor can be easily replaced by more complex drones in future work. The coordinate system of each drone D_i is located at its respective center of mass. The pose of a drone is defined by the vector $\hat{\mathbf{r}}_{Di}$. The vector \mathbf{b}_{Ri} points from the center of the coordinate system D_i to the cable attachment point of the drone. The orientation of each drone is described by

Cardan angles (axes x, y, z and angles $\alpha_{Di}, \beta_{Di}, \gamma_{Di}$), where, by definition, the angle around the drone-fixed z_{Di} -axis γ_{Di} is zero for all times. This leads to the transformation matrix ${}^{0D_i}T = T_x(\alpha_{Di})T_y(\beta_{Di})$. Since the derivation of the inverse model is the same for every drone, the index i is omitted from now on in this section. The Cardan angles α_D and β_D are related to the angular velocity ω_D by the so-called kinematic Cardan equation. For the representation of the angular velocity ω_D in the coordinate system of the drone, it holds

$$\begin{bmatrix} \dot{\alpha}_D \\ \dot{\beta}_D \\ \dot{\gamma}_D \end{bmatrix} = \frac{1}{\cos(\beta_D)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta_D) & 0 \\ -\sin(\beta_D) & 0 & \cos(\beta_D) \end{bmatrix} \begin{bmatrix} {}^D\omega_{Dx} \\ {}^D\omega_{Dy} \\ {}^D\omega_{Dz} \end{bmatrix}. \tag{10}$$

The drones must be able to carry the required forces at any time. To check whether a trajectory is feasible, the rotor angular velocities of all motors must be determined and evaluated for each drone at each time step. This implies that an inverse model of the system dynamics must be found based on Equations (4) and (8). For the drone modeling simplified equations are used. This implies the consideration of a drone as a rigid body using the thrust of each propeller as a direct input value. The thrust F_A and the drag τ_W of each propeller are each approximated by quadratic functions [50]

$$F_{A,p} = c_T \omega_p^2, \tag{11a}$$

$$\tau_{W,p} = c_W \omega_p^2, \tag{11b}$$

in terms of the propeller speed $\omega_p, p = M1, \dots, M4$. The parameter c_T is the thrust and the parameter c_W is the drag coefficient. These quantities depend mainly on the propeller used, but also on other factors such as the shape of the drone frame, which influences the air flow at the mounting position of the motor. The translational movement of a drone follows from Equation (4). Using the mass m_D , the acceleration due to the gravity g and the total thrust of all propellers F_A leads to

$$m_D \begin{bmatrix} \ddot{x}_D \\ \ddot{y}_D \\ \ddot{z}_D \end{bmatrix} = \begin{bmatrix} F_A \sin(\beta_D) \\ -F_A \cos(\beta_D) \sin(\alpha_D) \\ F_A \cos(\alpha_D) \cos(\beta_D) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_D g \end{bmatrix} - \begin{bmatrix} f_{Rx}^z \\ f_{Ry}^z \\ f_{Rz}^z \end{bmatrix}, \tag{12}$$

since $f_D^c = 0$ due to $b_{SD} = 0$. Equation (12) can directly be solved for the Cardan angles

$$\alpha_D = -\arctan\left(\frac{m_D \ddot{y}_D + f_{Ry}^z}{m_D(\ddot{z}_D + g) + f_{Rz}^z}\right), \tag{13a}$$

$$\beta_D = \arctan\left(\cos(\alpha_D) \frac{m_D \ddot{x}_D + f_{Rx}^z}{m_D(\ddot{z}_D + g) + f_{Rz}^z}\right), \tag{13b}$$

and the norm of the total thrust

$$F_A = \sqrt{(\ddot{x}_D m_D + f_{Rx}^z)^2 + (\ddot{y}_D m_D + f_{Ry}^z)^2 + ((\ddot{z}_D + g)m_D + f_{Rz}^z)^2}. \tag{14}$$

Under the assumption that the Coriolis and centrifugal forces can be neglected, the rotational movement of a drone follows from Equation (4) and is given by

$$\Theta_D \begin{bmatrix} {}^D\dot{\omega}_{Dx} \\ {}^D\dot{\omega}_{Dy} \\ {}^D\dot{\omega}_{Dz} \end{bmatrix} = \begin{bmatrix} c_T(\omega_{M2}^2 - \omega_{M4}^2)r_D \\ -c_T(\omega_{M1}^2 - \omega_{M3}^2)r_D \\ -c_W(\omega_{M1}^2 - \omega_{M2}^2 + \omega_{M3}^2 - \omega_{M4}^2) \end{bmatrix} + \begin{bmatrix} -\tau_{Rx} \\ \tau_{Ry} \\ 0 \end{bmatrix}. \tag{15a}$$

Note that this equation provides a direct link between the angular velocity of the drone ω_D in the drone-fixed coordinate system and the motor speeds. In Equation (15a), r_D is the radius of the drone, defined as the distance in the xy -plane of the drone between

the z-axis of the coordinate system \mathcal{D} and the axis of rotation of each propeller. The vector $\tau_{\mathbf{R}}$ is the constraint torque with its elements

$$\tau_{\mathbf{R}x} = b_{\mathbf{R}z} \left(f_{\mathbf{R}y}^z \cos(\alpha_{\mathcal{D}}) + f_{\mathbf{R}z}^z \sin(\alpha_{\mathcal{D}}) \right), \tag{15b}$$

$$\tau_{\mathbf{R}y} = b_{\mathbf{R}z} \left(f_{\mathbf{R}x}^z \cos(\beta_{\mathcal{D}}) + f_{\mathbf{R}y}^z \sin(\alpha_{\mathcal{D}}) \sin(\beta_{\mathcal{D}}) - f_{\mathbf{R}z}^z \cos(\alpha_{\mathcal{D}}) \sin(\beta_{\mathcal{D}}) \right). \tag{15c}$$

Equation (15) can now be solved for the angular velocity of each motor

$$\omega_p = \begin{bmatrix} \omega_{M1} \\ \omega_{M2} \\ \omega_{M3} \\ \omega_{M4} \end{bmatrix} = \frac{0.5}{\sqrt{r_{\mathcal{D}} c_{\mathcal{T}} c_{\mathcal{W}}}} \begin{bmatrix} \sqrt{r_{\mathcal{D}} c_{\mathcal{W}} F_A + 2c_{\mathcal{W}} \tau_{\mathbf{R}y} - 2\Theta_{\mathcal{D}y} c_{\mathcal{W}} \dot{\omega}_{\mathcal{D}y} - r_{\mathcal{D}} \Theta_{\mathcal{D}z} c_{\mathcal{T}} \dot{\omega}_{\mathcal{D}z}} \\ -\sqrt{r_{\mathcal{D}} c_{\mathcal{W}} F_A + 2c_{\mathcal{W}} \tau_{\mathbf{R}x} + 2\Theta_{\mathcal{D}x} c_{\mathcal{W}} \dot{\omega}_{\mathcal{D}x} + r_{\mathcal{D}} \Theta_{\mathcal{D}z} c_{\mathcal{T}} \dot{\omega}_{\mathcal{D}z}} \\ \sqrt{r_{\mathcal{D}} c_{\mathcal{W}} F_A - 2c_{\mathcal{W}} \tau_{\mathbf{R}y} + 2\Theta_{\mathcal{D}y} c_{\mathcal{W}} \dot{\omega}_{\mathcal{D}y} - r_{\mathcal{D}} \Theta_{\mathcal{D}z} c_{\mathcal{T}} \dot{\omega}_{\mathcal{D}z}} \\ -\sqrt{r_{\mathcal{D}} c_{\mathcal{W}} F_A - 2c_{\mathcal{W}} \tau_{\mathbf{R}x} - 2\Theta_{\mathcal{D}x} c_{\mathcal{W}} \dot{\omega}_{\mathcal{D}x} + r_{\mathcal{D}} \Theta_{\mathcal{D}z} c_{\mathcal{T}} \dot{\omega}_{\mathcal{D}z}} \end{bmatrix}. \tag{16}$$

It has to be stated that the inversion of the drone’s model only succeeds if $\mathbf{r}_{\mathbf{B}i}^{(n)} = \mathbf{r}_{\mathbf{D}i}^{(n)}$ holds for the n -th time derivative. Hence, the vector $\mathbf{b}_{\mathbf{R}D}$ should be as short as possible. With the knowledge of all cable forces $f_{\mathbf{R}i}$, the trajectory of the cable attachment point and the relating time derivatives can be calculated by

$$\mathbf{r}_{\mathbf{B}i}^{(n)} = \mathbf{r}_{\mathbf{R}i}^{(n)} + l_{0,i} \mathbf{v}_{\mathbf{R}i}^{(n)}. \tag{17}$$

In cases where two or more drones are used, Equation (8) leads to an underdetermined system of linear equations. As already mentioned, its solution can be found by

$$\mathbf{f}_{\mathbf{R}} = \left(\mathbf{P}^T \right)^+ \hat{\mathbf{f}}_{\mathbf{P}}^z + \left(\mathbf{E} - \left(\mathbf{P}^T \right)^+ \mathbf{P}^T \right) \lambda_0, \tag{18}$$

where $\left(\mathbf{P}^T \right)^+ \in \mathbb{R}^{3n_{\mathcal{D}} \times 6}$ is the Moore–Penrose inverse, $\mathbf{E}_{3n_{\mathcal{D}} \times 3n_{\mathcal{D}}}$ the identity matrix and $\lambda_0 \in \mathbb{R}^{3n_{\mathcal{D}} \times 1}$ an arbitrary vector. The time derivatives of $\mathbf{f}_{\mathbf{R}}$ can be calculated by differentiating first (8) with respect to time using the Leibniz rule

$$\hat{\mathbf{f}}_{\mathbf{P}}^{z(n)} = \sum_{k=0}^n \binom{n}{k} \mathbf{P}^{T(k)}(x) \mathbf{f}_{\mathbf{R}}^{(n-k)}(x), \tag{19}$$

and solving this equations analogous to the procedure of (18) afterwards, concerning that the vectors λ_i for the time derivatives are chosen equal to zero.

For the modeling of the drone dynamics, it is assumed that the coordinate axes of \mathcal{E} are aligned with the platform’s principal axes of inertia. This also holds for the homogeneous masonry unit. Assuming that a masonry unit is always gripped centrally, this yields $\mathbf{b}_{\mathcal{S}} = [0 \ 0 \ b_{\mathcal{S}z}]^T$ and $\Theta_{\mathcal{P}} = \text{diag}(\Theta_{\mathcal{P}xx}, \Theta_{\mathcal{P}yy}, \Theta_{\mathcal{P}zz})$. Note that $\mathbf{b}_{\mathcal{S}}$ and $\Theta_{\mathcal{P}}$ have to be recalculated for every trajectory. Additionally, by default the platform performs only rotations around the body-fixed $z_{\mathcal{P}}$ -axis. Thus, the orientation of the platform can be described by the matrix $T_{\mathcal{P}}(\gamma_{\mathcal{P}})$. That means that there is the direct relation $\omega_{\mathcal{P}} = \dot{\gamma}_{\mathcal{P}}$ between the angular velocity and the time derivative of the platform angle around the z -axis. This simplifies Equation (6) into

$$\hat{\mathbf{f}}_{\mathbf{P}}^z = \hat{\mathbf{M}}_{\mathcal{P}} \ddot{\mathbf{r}}_{\mathcal{P}} - \hat{\mathbf{f}}_{\mathcal{P}}^c, \quad \text{with } \hat{\mathbf{f}}_{\mathcal{P}}^c = \mathbf{0}. \tag{20}$$

The necessary time derivatives of Equation (20) in Equation (19) can be calculated by

$$\hat{\mathbf{f}}_{\mathbf{P}}^{z(n)} = \hat{\mathbf{M}}_{\mathcal{P}} \hat{\mathbf{r}}_{\mathcal{P}}^{(n+2)}, \tag{21}$$

since the product $\dot{M}_P \ddot{r}_P$ of the time derivative of the mass matrix

$$\dot{M}_P = \begin{bmatrix} \mathbf{0} & m_P \dot{\hat{b}}_S^T \\ m_P \dot{\hat{b}}_S & \dot{\Theta}_P \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\omega}_P \Theta_P - \Theta_P \tilde{\omega}_P \end{bmatrix} \quad (22)$$

and the acceleration vector $\ddot{r}_P = [\ddot{r}_P \ 0 \ 0 \ \dot{\omega}_{Pz}]^T$ is always zero. This follows because

$$\tilde{\omega}_P \Theta_P - \Theta_P \tilde{\omega}_P = \begin{bmatrix} 0 & \omega_{Pz}(\Theta_{Pxx} - \Theta_{Pyy}) & 0 \\ \omega_{Pz}(\Theta_{Pxx} - \Theta_{Pyy}) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (23)$$

Moreover, the time derivatives of the vector of the applied force $\hat{f}_P^{e(n)}$ are always zero, since the constant gravitational force is the only applied force which is taken into account.

Note that the calculation of the propeller speeds ω_p in Equation (16) depends on $\dot{\omega}_D$. Solving Equation (10) for ω_D states that the calculation of $\dot{\omega}_D$ requires the second time derivatives $\ddot{\alpha}_D$ and $\ddot{\beta}_D$. Therefore, Equation (13) must be differentiated twice, which means that the fourth time derivative $r_D^{(4)} = r_B^{(4)}$ has to be determined. Furthermore, it follows from Equations (8) and (17) that $v_B^{(4)}$ and $f_R^{(4)}$ must also be known. Equation (18) states that this requires knowledge of the fourth time derivative of the constraint force \hat{f}_P^z . Finally, it follows from Equation (21) that the sixth time derivative of the platform pose \hat{r}_P is needed. For each trajectory and its six time derivatives, the boundary conditions at the beginning and end of the trajectory must be defined (i.e., two times a set of one plus six conditions) to ensure continuity, resulting in a total of 14 boundary conditions per DOF. Using polynomials for the trajectory, a polynomial of the 13th order provides 14 coefficients. Solving a system of these 14 unknown coefficients and the 14 boundary conditions thus defines the trajectory meeting all conditions at the start, the end and in all time derivatives. An example of the use of a polynomial function for a trajectory can be found in [51].

Equation (16) leads to the cost function

$$\mathcal{V}_\omega = \frac{1}{n_s n_D n_M} \left(\sum_{k=1}^{n_s} \sum_{i=1}^{n_D} \sum_{j=1}^{n_M} \frac{\omega_{M,i,j}(k) - \omega_{M,\min}}{\omega_{M,\max} - \omega_{M,\min}} \right)^3. \quad (24)$$

If the actual speed ω_M is above the maximum speed $\omega_{M,\max}$ or below minimum speed $\omega_{M,\min}$, the penalty term \mathcal{P}_ω is applied and \mathcal{V}_ω is not assigned.

3.3. Modeling of the Cable Robot’s Winch Dynamics, Power and Energy Consumption

Winch dynamics: The moment of inertia of the drive train, reduced to the winch side, is computed as

$$J = J_W + J_C + J_M i_G^2 + J_G i_G^2, \quad (25)$$

by taking into account the moment of inertia of the winch J_W , the clutch J_C , the motor shaft J_M and the gearbox J_G with its gear ratio i_G . Using the winch drum radius r_W and the cable acceleration \ddot{q}_R , one finds for the total motor torque of the i -th cable

$$\tau_{t,i} = \frac{f_{R,i} r_W}{i_G \eta_G} + \frac{J \ddot{q}_R i}{i_G \eta_G r_W}. \quad (26)$$

The friction of the overall drive train is modeled by an overall efficiency η_G (the efficiency η_G can be derived from the manufacturer’s catalog data at full load of the gearbox). As the drive train torques are limited by $\tau_{t,\max}$, a penalty $\mathcal{P}_{\text{torque}}$ is set in case $\tau_{t,i}$ exceeds the capabilities. Note that, in this work, $\mathcal{V}_{\text{torque}} = 0$.

Power and energy consumption: The used electric motors are highly efficient permanent magnet synchronous motors. The detailed electric modeling of the motor is simplified

by a power balance approach. The mechanical winch power per cable at time step k can be computed as

$$P_{M,i}(k) = \tau_{t,i}(k)\omega_{W,i}(k) , \tag{27}$$

where $\omega_{W,i}$ is the angular velocity of the i -th winch drum. Considering the motor’s phase to phase winding resistance R_{PP} and its motor torque constant k_t from the specifications in the data sheet, the copper losses

$$P_{C,i} = \frac{3}{2} \left(\frac{\tau_{t,i}}{k_t} \right)^2 R_{PP} \tag{28}$$

can be computed [52]. Note that the motor has three phases and is star connected. Taking into account also the inverter efficiency η_{ESC} and the recuperation efficiency η_R leads to the total, needed or recuperated winch power

$$P_{W,i} = \begin{cases} (P_{M,i}\eta_G + P_{C,i}) \eta_{ESC} \eta_R & \text{for } P_{M,i} + P_{C,i} < 0 \\ (P_{M,i} \frac{1}{\eta_G} + P_{C,i}) \frac{1}{\eta_{ESC}} & \text{for } P_{M,i} + P_{C,i} \geq 0 \end{cases} . \tag{29}$$

For every time step, an overall power

$$P(k) = \sum_{i=1}^{n_c} P_{W,i}(k) \tag{30}$$

can be computed. In case $P(k)$ exceeds the maximum power P_{max} provided by the power source at any time step k , a penalty \mathcal{P}_{power} is set. From the power requirements over time, the energy consumption is approximated using the trapezoidal rule [53]

$$E_{total} = \sum_{k=2}^{n_s} \frac{P(k) + P(k-1)}{2} \Delta t , \tag{31}$$

where $\Delta t = \frac{t_{total}}{n_s}$ is the step size of a time step. If used, the total energy consumption defines its costs $\mathcal{V}_{energy} = E_{total}$. Later on E_{total} is used to compare both systems. Notably, neither \mathcal{V}_{power} , \mathcal{V}_{energy} nor \mathcal{P}_{energy} is employed within the optimization and is thus set to zero.

The considered CDPR has eight cables in each corner of the frame, which are guided into the workspace using pivoting deflection pulleys. The four lower pulleys (see Equation (47), ${}^B r_{B1}, {}^B r_{B2}, {}^B r_{B5}, {}^B r_{B6}$) can be moved vertically. They are lifted as soon as a layer of masonry units has been finished. The movement needed is done between trajectories 14 and 15. Therefore, it is assumed that these pulleys are attached to slides that may move along vertical rails, powered by external drives. The dynamic effects and power consumption is neglected. Note that the details on the modeling of these effects can be found in [48]. The actuated sliding motion of the pulleys under cable tension and alongside a trajectory has already been demonstrated [54].

3.4. Modeling of Collisions

The following collision detection formalisms are based on geometric considerations. The necessary fundamentals can be found in [55].

Self-collision of the robot and collisions of the robot with its environment: For collision avoidance, all relevant objects such as drones, end effector, payload and static obstacles are represented by cuboid geometries called Axis-Aligned Bounding Boxes (AABB) [56]. As these boxes are aligned with the three axes of the inertial system, both distance computations as well as collision checks are massively simplified. In case relevant rotations of the bodies involved are planned, the AABBs can be replaced by so-called Oriented Bounding Boxes (OBB) [56] for the price of slightly increased computation costs.

Collision on AABBs can be trivially checked using the Separating Axis Theorem [56]. Two convex objects can not collide if a separating (hyper-)plane and its normal—the

separating axis—exist. Accordingly, if in at least one axis no overlap of the boxes can be found—which is trivial to check per-axis of the inertial system—the AABBs do not collide. Given two AABBs 1 and 2, their center coordinates $r_1 = [x_1, y_1, z_1]^T$ and $r_2 = [x_2, y_2, z_2]^T$ as well as their halfwidth extends e_{1x}, e_{1y}, e_{1z} and e_{2x}, e_{2y}, e_{2z} , respectively, the following distances can be computed

$$\begin{aligned} d_x &= \text{abs}(x_1 - x_2) - (e_{1x} + e_{2x}), \\ d_y &= \text{abs}(y_1 - y_2) - (e_{1y} + e_{2y}), \\ d_z &= \text{abs}(z_1 - z_2) - (e_{1z} + e_{2z}). \end{aligned} \tag{32}$$

If all distances d_x, d_y, d_z become negative at the same time, a collision occurs. This allows to check the potential collisions $\chi \in \{DD, \text{ObEE}, \text{ObD}\}$ where, in case of collision, $\mathcal{P}_{DD}, \mathcal{P}_{\text{ObEE}}$ or \mathcal{P}_{ObD} , respectively, receives penalty terms.

Furthermore, the distance of the AABBs is checked to support the convergence of the optimizer. Thus, the costs of $\mathcal{V}_{DD}, \mathcal{V}_{\text{ObEE}}$ and \mathcal{V}_{ObD} are calculated by observing the smallest distance

$$d_{ij} = \sqrt{d_x^2 + d_y^2 + d_z^2} \tag{33}$$

of each two AABBs, which generally is the distance of the two closest corners. The costs

$$\mathcal{V}_\chi = \frac{1}{n_s n_{oo}} \left(\sum_{i=1}^{n_o-1} \sum_{j=1+i}^{n_o} \sum_{k=1}^{n_s} 1 - \frac{d_{ij}(k)}{d_{\min}} \right), \quad n_{oo} = \binom{n_o}{2}, \quad \chi \in \{DD, \text{ObEE}, \text{ObD}\}. \tag{34}$$

are respectively assigned as long as the condition $0 \leq d_{ij}(k) \leq d_{\min}$ is met. Distances of two AABBs larger than a predefined distance d_{\min} are treated as uncritical. In this case ($d_{ij}(k) > d_{\min}$), $\mathcal{V}_{DD}, \mathcal{V}_{\text{ObEE}}$ and \mathcal{V}_{ObD} are set to zero to avoid negative cost values. Note that the binomial coefficient n_{oo} equals the number of collision checks for n_o obstacles.

Self-collision of cables: The method for detecting self-collisions of cables takes advantage of the fact that the end effector follows a continuous path. Whether during the last time step Δt a collision might have happened or not, can be checked by simple geometric considerations using the given quantities in Figure 4a.

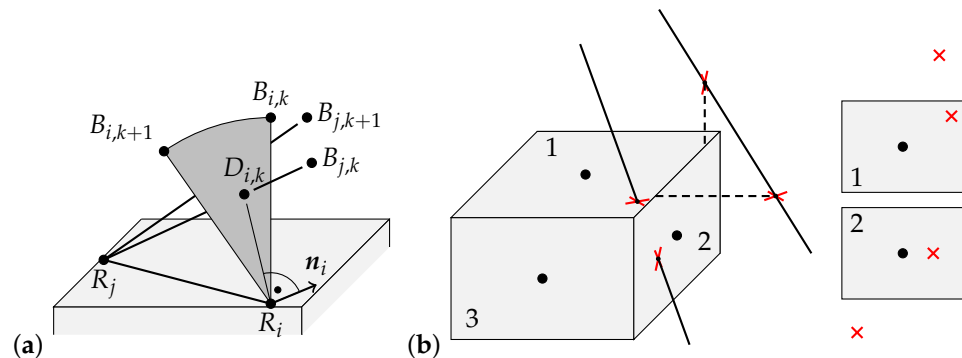


Figure 4. Schematic illustration of the method for collision detection between a cable and (a) another cable and (b) an obstacle.

Accordingly, it is assumed that the time interval is chosen sufficiently small so that the motion can be considered as linear between two time steps. Two cables S_i and S_j are shown in the platform-fixed coordinate system \mathcal{E} for the two successive time steps t_k and t_{k+1} . In this coordinate system, the vectors p_{R_i} and p_{R_j} to the end effector’s cable attachment points $R_i = R_{i,k} = R_{i,k+1}$ and $R_j = R_{j,k} = R_{j,k+1}$ are constant. However, the cable attachment points at the drones B_i and B_j with their associated vectors r_{B_i} and r_{B_j} move relatively to the end effector within the time step on an circular arc with the radius l_0 . Thereby, the cable S_i sweeps out the sector area K_i defined by the three points $R_i, B_{i,k}$ and $B_{i,k+1}$ while the

cable S_j sweeps out the sector area K_j defined by the three points R_j , $B_{j,k}$ and $B_{j,k+1}$. Thus, for a collision test, the four cable vectors

$$l_{i,k} = r_{B_{i,k}} - r_{R_i}, \quad l_{i,k+1} = r_{B_{i,k+1}} - r_{R_i}, \quad l_{j,k} = r_{B_{j,k}} - r_{R_j}, \quad l_{j,k+1} = r_{B_{j,k+1}} - r_{R_j} \quad (35)$$

must be defined. Given that the two sector areas are not parallel to each other, a point of intersection between the sector area K_i and the cable S_j as well as between the sector area K_j and the cable S_i can always be calculated for each time step. A collision can only be entirely excluded if the calculated points of intersection $D_{i,k}$ and $D_{i,k+1}$ between the sector area K_i and the cable S_j as well as the calculated points of intersection $D_{j,k}$ and $D_{j,k+1}$ between the sector area K_j and the cable S_i are all outside the sector areas.

For the actual collision check, the two sector areas are treated as planes, expressed by their normal form $(x - p)^T n = 0$. Here, x is the vector to an arbitrary point on the plane, p is the vector to the cable attachment point of the considered sector area and n is the normal vector of the plane. In a first step, the intersection points $D_{i,k}$ and $D_{i,k+1}$ between the plane K_i and the cable S_j expressed by the vector l_j are calculated separately for the two time steps t_k and t_{k+1} . For this purpose, the vector x in the normal form of the plane is successively substituted by the parametric equation of the cables. The vector r_{R_i} is chosen as p . This yields the two vectors

$$d_{i,k} = r_{R_j} + s_{j,k} l_{j,k}, \quad \text{with } s_{j,k} = \frac{(r_{R_i} - r_{R_j})^T n_i}{l_{j,k}^T n_i}, \quad (36a)$$

$$d_{i,k+1} = r_{R_j} + s_{j,k+1} l_{j,k+1}, \quad \text{with } s_{j,k+1} = \frac{(r_{R_i} - r_{R_j})^T n_i}{l_{j,k+1}^T n_i} \quad (36b)$$

of the intersection points D_i . The first condition

$$C_{1i,k} : (0 < s_{j,k} \leq 1), \quad C_{1i,k+1} : (0 < s_{j,k+1} \leq 1) \quad (37)$$

checks separately for each time step whether the point of intersection is within the free cable section between the last pulley on frame side and the platform. Otherwise a collision can be excluded. If one of the conditions $C_{1i,k}$ or $C_{1i,k+1}$ is fulfilled, the second step is to check if the calculated points of intersection are inside the sector area. For this, the sector area is defined by its parametric equation $x_i = p_{R_i} + s_{i,k} l_{i,k} + s_{i,k+1} l_{i,k+1}$, with the affine coordinates of a point $s_{i,k}$ and $s_{i,k+1}$. Inserting the points of intersection $d_{i,k}$ and $d_{i,k+1}$ from Equation (36) in the parametric equation leads to their position vectors

$$s_{D,i,k} = S_i^+ (d_{i,k} - r_{R_i}) \quad s_{D,i,k} \in \mathbb{R}^{2 \times 1}, \quad (38a)$$

$$s_{D,i,k+1} = S_i^+ (d_{i,k+1} - r_{R_i}) \quad s_{D,i,k+1} \in \mathbb{R}^{2 \times 1}. \quad (38b)$$

Herein, the matrix S_i^+ is the pseudo inverse of the matrix $S_i = [l_{i,k} \quad l_{i,k+1}]$. The intersection point is within the sector area, provided that for the two timesteps the conditions

$$C_{2i,k} : ((s_{D,i,k}(1) > 0) \wedge (s_{D,i,k}(2) > 0)), \quad C_{3i,k} : \|S_i s_{D,i,k}\|_2 \leq l_0 \quad (39)$$

$$C_{2i,k+1} : ((s_{D,i,k+1}(1) > 0) \wedge (s_{D,i,k+1}(2) > 0)), \quad C_{3i,k+1} : \|S_i s_{D,i,k+1}\|_2 \leq l_0 \quad (40)$$

are true. This procedure must be repeated for the sector area K_j and the cable S_i . This results in the conditions $C_{1j,k}$, $C_{2j,k}$ and $C_{3j,k}$ for the time step t_k and $C_{1j,k+1}$, $C_{2j,k+1}$ and $C_{3j,k+1}$ for the time step t_{k+1} . The overall condition can be defined as

$$C_{ges} = (C_{1i,k} \wedge C_{2i,k} \wedge C_{3i,k}) \vee (C_{1i,k+1} \wedge C_{2i,k+1} \wedge C_{3i,k+1}) \vee (C_{1j,k} \wedge C_{2j,k} \wedge C_{3j,k}) \vee (C_{1j,k+1} \wedge C_{2j,k+1} \wedge C_{3j,k+1}). \quad (41)$$

Finally it can be stated that

- a collision can be excluded, if $C_{ges} = 0$ and that
- a collision has happened within the chosen time step, if $C_{ges} = 1$.

Provided that a collision took place, it can be useful to repeat the collision check with a reduced step size. In the case of a cable collision, the penalty \mathcal{P}_{CaCa} is added. For cable–cable collisions, no costs are calculated, which is why $\mathcal{V}_{CaCa} = 0$. Inspired by Gouttefarde [57] and due to its simplicity, an approach easy to implement was chosen. More sophisticated approaches can be found in [58–61].

Cable–obstacle collisions: For a collision check between a cable and an obstacle according to Figure 4b, all $f = 1, \dots, 6$ faces of a AABB are checked individually for a collision with every cable i . The vector \mathbf{c}_f points to the center of each face, which can be derived using the halfwidth extends $\mathbf{e}_f = [e_{fx} \ e_{fy} \ e_{fz}]^T$ of the AABB. Analogous to the procedure for a collision of type $\chi = CaCa$, each face of the AABB is defined as a plane in its normal form $(\mathbf{x}_i - \mathbf{c}_f)^T \mathbf{n}_f = 0$. The cable under consideration S_i is again expressed by its vector \mathbf{l}_i . With these two definitions, each point of intersection

$$\mathbf{d}_{i,f} = \mathbf{r}_{Ri} + s_{i,f} \mathbf{l}_i, \quad \text{with } s_{i,f} = \frac{(\mathbf{c}_f - \mathbf{r}_{Ri})^T \mathbf{n}_f}{\mathbf{l}_i^T \mathbf{n}_f}, \quad f = 1, \dots, 6, \quad i = 1, \dots, n_c \quad (42)$$

is calculated for all n_c cables. From the center of each face, the vector

$$\mathbf{q}_{i,f} = \begin{bmatrix} q_{i,f,x} \\ q_{i,f,y} \\ q_{i,f,z} \end{bmatrix} = (\mathbf{d}_{i,f} - \mathbf{c}_f) \quad (43)$$

points to the point of intersection of the considered cable and face. Provided that the conditions

$$C_1 : \left((|q_{i,f,x}| \leq e_{fx}) \wedge (|q_{i,f,y}| \leq e_{fy}) \wedge (|q_{i,f,z}| \leq e_{fz}) \right), \quad (44a)$$

$$C_2 : \left\| \mathbf{d}_{i,f} - \mathbf{r}_{Ri} \right\|_2 \leq l_0 \quad (44b)$$

are both satisfied, there is a collision and the procedure is terminated with the penalty term \mathcal{P}_{CaOb} for the considered cable. In contrast to the cable–cable collision, for the cable–obstacle collision a cost function is defined. For this purpose, for each cable $i = 1, \dots, n_c$ the closest distance to one of the surfaces of all objects $j = 1, \dots, n_o$ is defined by

$$d_{ij} = \min_{\forall f} \left(\left\| \mathbf{d}_{i,j,f} - \mathbf{c}_{i,j,f} \right\|_2 \right) \quad (45)$$

This leads to the total cable–obstacle collision costs for one time step

$$\mathcal{V}_{CaOb} = \frac{1}{n_c n_s n_o} \left(\sum_{i=1}^{n_c} \sum_{j=1}^{n_o} \sum_{k=1}^{n_s} 1 - \frac{d_{ij}(k)}{d_{\min}} \right). \quad (46)$$

Here, analogous to Equation (34), the value of d_{\min} specifies a boundary above which the costs are zero.

4. Discussion of the Optimization Results

The following scenarios under consideration deal with the automatic construction of a building consisting of 1720 calcium silicate units. These mainly include calcium silicate masonry units of the type 6DF [62,63], as well as flat lintels and masonry units to compensate height differences. Note that, for all units, the optimization has ended penalty-free for both systems, i.e., all undesired events could be avoided and all final optimization runs terminated with valid trajectories.

In the first scenario, a drone system with three quadcopters as shown in Figure 1b is used to transport the payload past some already erected buildings to the actual construction site. In this case, the optimizer is forced to plan a path which avoids collisions. The scenario is used to verify the functioning of the implemented algorithms. The drone system used in this first scenario has the parameters listed in Table 2. Notably, with the given drone system, the platform is kinematically underconstrained [34], which means that the system might oscillate in case of disturbances. Applicable control approaches are subject to future work, where e.g., flatness-based approaches are promising [64].

To identify and compare the advantages and disadvantages of each robotic system, the second scenario deals with the optimization of the entire construction process for all 1720 masonry units for this drone system and the cable robot presented in Figure 1a. For the optimization process, all trajectories are employed as given in Table 1. Note that the mass of the platform is subject to instantaneous changes after gripping the masonry unit between trajectories 3 and 4 and after placing the masonry unit between trajectories 10 and 11. This inevitably leads to discontinuities, e.g., in cable forces or propeller speeds. Such discontinuities might as well lead to theoretically infinite accelerations, e.g., at the motors of the drones. However, since the simulation framework is using an open loop physical model and calculates each trajectory independently, this does not lead to problems within the simulation. The effects of this short time period were assumed to be negligible in terms of energy consumption and overall construction time. Thus, a modeling of such effects was not included within this work. Future work includes the modeling of the contact forces when gripping or releasing a masonry unit.

Table 2. Drone system parameters.

Parameter	Value
Number and type of drones	$n_D = 3$ quadcopters
Platform mass	$m_P = 20$ kg
Drone mass	$m_D = 14.26$ kg
Masonry unit mass	$m_B = 10.7$ kg– 53.7 kg
Length of the i -th cable	$l_{0,i} = 5.7$ m
Vector from the drone's center of gravity to the attachment point of the cable	$\mathbf{b}_{R_i} = [0 \text{ m} \quad 0 \text{ m} \quad -0.04 \text{ m}]^T$
Tensor of inertia of the drone	$\Theta_D \approx \text{diag}(0.33 \text{ kgm}^2, 0.33 \text{ kgm}^2, 0.78 \text{ kgm}^2)$
Tensor of inertia of the platform	$\Theta_P \approx \text{diag}(0.2 \text{ kgm}^2, 0.2 \text{ kgm}^2, 0.2 \text{ kgm}^2)$
Thrust coefficient	$c_T = 7.243 \cdot 10^{-4} \text{ N s}^2 / \text{rad}^2$
Drag coefficient	$c_W = 3.137 \cdot 10^{-5} \text{ N m s}^2 / \text{rad}^2$
Rotor speed	$\omega_{M,\max} = 560$ rad/s
Vector to the cable attachment points of the platform	$\mathbf{p}_{R1} = [0.34 \text{ m} \quad -0.1525 \text{ m} \quad 0.6 \text{ m}]^T$ $\mathbf{p}_{R2} = [0 \text{ m} \quad 0.305 \text{ m} \quad 0.6 \text{ m}]^T$ $\mathbf{p}_{R3} = [-0.34 \text{ m} \quad -0.1525 \text{ m} \quad 0.6 \text{ m}]^T$

The cable robot’s parameters are given in [48]. The most important ones are

$$\begin{aligned}
 [{}^{\mathcal{E}}\mathbf{p}_{R1} \dots {}^{\mathcal{E}}\mathbf{p}_{R8}] &= \begin{bmatrix} 0.51 & -0.51 & 0.51 & -0.51 & 0.51 & -0.51 & 0.51 & -0.51 \\ -0.0315 & -0.0315 & -0.445 & -0.445 & 0.0315 & 0.0315 & 0.445 & 0.445 \\ 0.714 & 0.714 & 0.324 & 0.324 & 0.714 & 0.714 & 0.324 & 0.324 \end{bmatrix} \text{ m,} \\
 [{}^{\mathcal{B}}\mathbf{r}_{B1} \dots {}^{\mathcal{B}}\mathbf{r}_{B8}] &= \begin{bmatrix} 4.8 & -4.8 & 4.33 & -4.33 & 4.8 & -4.8 & 4.33 & -4.33 \\ -6.996 & -6.996 & -7.096 & -7.096 & 6.996 & 6.996 & 7.096 & 7.096 \\ 0.5 & 0.5 & 5.1875 & 5.1875 & 0.5 & 0.5 & 5.1875 & 5.1875 \end{bmatrix} \text{ m,} \\
 {}^{\mathcal{B}}\mathbf{b}_{SP} &= [-0.0078 \ -0.0074 \ 0.4059]^T \text{ m,} \\
 {}^{\mathcal{B}}\boldsymbol{\theta}_S &= \begin{bmatrix} 11.0424 & 0.3203 & 0.4189 \\ 0.3203 & 10.8053 & 0.3697 \\ 0.4189 & 0.3697 & 9.8329 \end{bmatrix} \text{ kg m}^2. \tag{47}
 \end{aligned}$$

The end effectors of both systems differ in their ability to rotate the masonry unit. In the case of the drone system, the rotation comes from the absolute positioning of the drones. In comparison, the end effector of the cable robot can actively rotate a masonry unit. For this reason, the end effector of the drone system can be built much lighter since the rotation unit, using a powerful electric motor, heavy bearings and an extra frame [9], is omitted. Moreover, the frame of the end effector for the drone can be built using more lightweight elements since the expected cable forces are lower in comparison to the CDPR and the cables are not tensed against each other. Additionally, five cable attachment points and five cable force sensors are not needed. Therefore, it is assumed that the weight $m_{EE} = 100$ kg of the end effector of the cable robot can be reduced to $m_{EE} = 20$ kg for the drone system.

Notably, in both scenarios, the 6th trajectory—which is basically the motion across the site—of the 1000th masonry unit is discussed in more detail with its starting position A and its target position B . Apart from the lateral movement from its starting position $\mathbf{r}_{P,A}$ to its target position $\mathbf{r}_{P,B}$, the masonry unit is additionally rotated by $\varphi = -90^\circ$ around the z_P -axis. Hereby, the target position and the rotation angle result from the masonry laying plan. Note that cable angles α_S, β_S are used for evaluation. These angles are defined as the Cardan angles analog to Section 3.2, such that the initial unit cable vector $\mathbf{v}_{R0} = [0 \ 0 \ 1]^T$ is mapped to \mathbf{v}_{Ri} .

First scenario: For the first scenario, the 6th trajectory starts at point A with $\mathbf{r}_{P,A} = [30 \text{ m} \ -6 \text{ m} \ 2.45 \text{ m}]^T$ and ends at point B with $\mathbf{r}_{P,B} = [2.8625 \text{ m} \ 3.9 \text{ m} \ 2.45 \text{ m}]^T$ passing two additional knots S_1 and S_2 . These knots with their vectors $\mathbf{r}_{P,S1}$ and $\mathbf{r}_{P,S2}$ are optimized as well as the platform’s velocity and acceleration at these points. The relative position of the drones to each other is influenced by the definition of the vector λ_0 in Equation (18). The choice

$$\lambda_0 = \begin{bmatrix} \lambda_{0,1} \\ \lambda_{0,2} \\ \lambda_{0,3} \end{bmatrix} \lambda_S, \quad \lambda_{0,1} = \begin{bmatrix} \sin(60^\circ) \\ -\cos(60^\circ) \\ 1 \end{bmatrix}, \quad \lambda_{0,2} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \lambda_{0,3} = \begin{bmatrix} -\sin(60^\circ) \\ -\cos(60^\circ) \\ 1 \end{bmatrix}. \tag{48}$$

results in solutions where the drones are as close as possible to an angle of 120° around the z_P -axis to each other. The scalar value $\lambda_S \in \mathbb{R}$ can be used to adjust the spreading between the drones, since the force f_R calculated with Equation (18) is the one that has the smallest distance to λ_0 with respect to Euclidean norm. Therefore, an increasing value λ_S results in a greater distance between the drones.

The optimization is done using the introduced Particle Swarm Optimizer of MATLAB® with its default settings [65] and the parameters of the cost functions as shown in Table 3.

Note that the comparatively high value of \mathcal{W}_ω results from the power of three in Equation (24) and the need for small drone angles α_D and β_D to meet the assumption $\mathbf{r}_{Bi}^{(n)} \approx \mathbf{r}_{Di}^{(n)}$. Figure 5 shows the optimization result for trajectories 4 to 10.

Table 3. Optimization parameters for the drone system.

Cost Function	Cost	Weight	Penalty	Limits
Time	1	$1 \frac{1}{s}$	-	-
Propeller speed	1	21,952	10,000	$5 \dots 560 \text{ rad/s}$
DD	1	1	$1 \cdot 10^5$	$d_{\min} = 1 \text{ m}$
ObEE	1	1	1000	$d_{\min} = 0.1 \text{ m}$
ObD	1	0	$1 \cdot 10^6$	$d_{\min} = 0.1 \text{ m}$
CaCa	1	0	$1 \cdot 10^6$	$d_{\min} = 0.1 \text{ m}$
CaOb	1	1	$1 \cdot 10^6$	$d_{\min} = 0.1 \text{ m}$
Position end effector	1	0	1000	$r_{Pz} = 0 \dots 4.5 \text{ m}$
Velocity end effector	1	0	1000	$ v_{\max} = 3 \frac{\text{m}}{\text{s}}$
Acceleration end effector	1	0	1000	$ a_{\max} = 5 \frac{\text{m}}{\text{s}^2}$

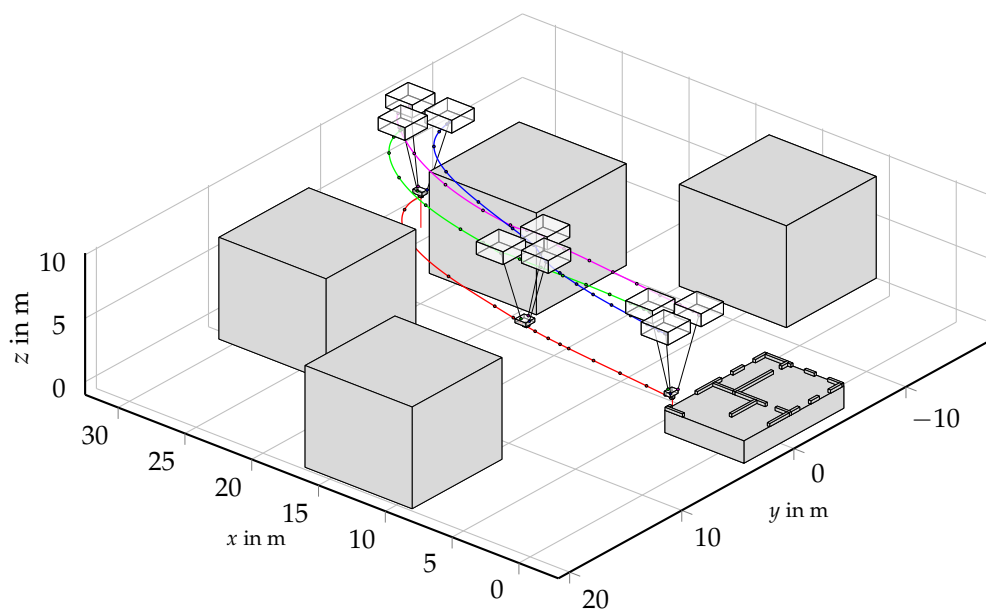


Figure 5. Optimization result for the 1000th masonry unit for scenario 1, using a drone system.

As desired, the optimized trajectories avoid all obstacles.

For the 6th trajectory, the transition time is reduced by the optimizer from initially $t_t = 50 \text{ s}$ to $t_t \approx 27.02 \text{ s}$. The platform is with $\hat{r}_{P,A}^{(i)} = \hat{r}_{P,B}^{(i)} = \mathbf{0}$ in perfect rest at the start and target position. The optimizer changed the initial position, velocity and acceleration vectors of the platform at the additional knot S1 and S2 to the vectors given in Table 4. To obtain a better impression of the actual platform and drone speeds, the calculated position of the drone’s cable attachment points are additionally highlighted at constant time intervals. For the rotation of the masonry unit, the whole end effector has to be rotated. Therefore, the drones must change their absolute position to each other in a coordinated manner.

This fact can be seen in Figure 5 by the intersections of their flight paths. Moreover, the relative alignment of the drones at approximately 120° is clearly visible. After the optimization, the spreading factor has the value $\lambda_S \approx 33.06$. This value is mainly affected by two cost functions. The costs of the rotor speeds \mathcal{V}_ω are increasing as λ_S becomes larger and, thus, the optimizer tries to achieve the smallest possible λ_S . The cost function \mathcal{V}_{DD} has the opposite effect and increases when λ_S decreases, as the AABBs get closer. Thus, λ_S can be influenced by the weights of \mathcal{W}_ω and \mathcal{W}_{DD} .

Table 4. Optimizationresult for the 1000th masonry unit for scenario 1 at the knots S1 and S2 of trajectory 6.

Knot	Position	Velocity	Acceleration
Knot S1 at $t_{S1} \approx 10.81$ s	$r_{P,S1} \approx \begin{bmatrix} 16.32 \\ 2.08 \\ 2.45 \end{bmatrix}$ m	$\dot{r}_{P,S1} \approx \begin{bmatrix} -0.96 \\ 0.16 \\ -0.08 \end{bmatrix}$ $\frac{m}{s}$	$\ddot{r}_{P,S1} \approx \begin{bmatrix} 0.861 \\ -0.10 \\ 0.10 \end{bmatrix}$ $\frac{m}{s^2}$
Knot S2 at $t_{S2} \approx 16.21$ s	$r_{P,S2} \approx \begin{bmatrix} 11.57 \\ 2.82 \\ 2.33 \end{bmatrix}$ m	$\dot{r}_{P,S2} \approx \begin{bmatrix} -1.26 \\ 0.17 \\ 0.03 \end{bmatrix}$ $\frac{m}{s}$	$\ddot{r}_{P,S2} \approx \begin{bmatrix} -0.03 \\ -0.01 \\ 0.00 \end{bmatrix}$ $\frac{m}{s^2}$

The associated cable and drone angles are shown in Figure 6a–c.

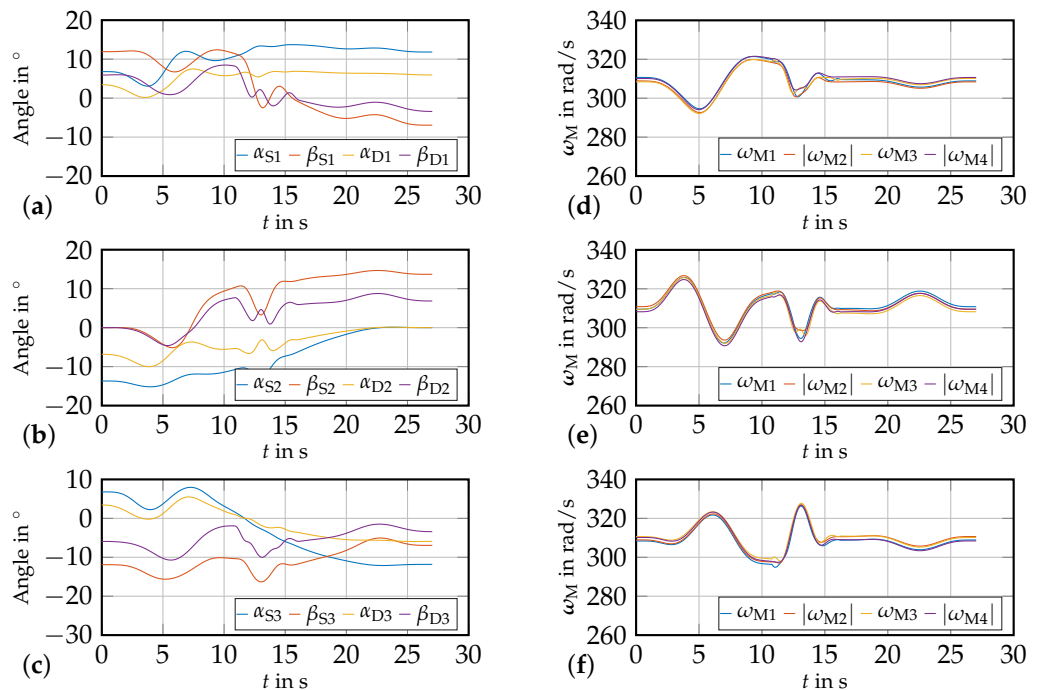


Figure 6. Drone and cable angles in (a–c) and rotor speeds (d–f) for the optimized scenario 1.

As an example, the curves for the drone D2 are considered in more detail at the start and target position. From Equation (48), it follows that at the starting point, the cable vector of this drone lies within a plane parallel to the yz -plane of the inertial frame, pointing in positive y - and z -direction. This requires a single rotation around the x -axis of the inertial frame with $\alpha_{S2} < 0^\circ$ and $\beta_{S2} = 0^\circ$. To achieve a stable system behavior while hovering, the drone must be tilted less than the cable and, therefore, $|\alpha_{D2}| < |\alpha_{S2}|$ must hold. Both conditions are confirmed by the angles calculated by the optimizer. At the target position, the drone D2 is oriented in such way that the cable lies within a plane parallel to the xz -plane of the inertial frame, pointing in positive x - and z -direction. This requires a single rotation with $\beta_{S2} > 0^\circ$ and $\alpha_{S2} = 0^\circ$. Furthermore, $|\beta_{D2}| < |\beta_{S2}|$ must hold. These considerations are also confirmed by the results in Figure 6b. The same procedure can be used to check the other cable and drone angles at the start and target positions. During the trajectory, maximum angles of $\alpha_{S,D}, \beta_{S,D} < 15^\circ$ occur. The differences between the cable and drone angles are $\Delta\alpha_{S,D} = |\alpha_S - \alpha_D| < 6^\circ$ and $\Delta\beta_{S,D} = |\beta_S - \beta_D| < 6^\circ$. The results of the required rotor speeds are shown in the Figure 6d–f separately for each drone. It can be seen that, due to the arrangement of the cables at the end effector, all drones must carry the same load in stationary case. To reach $\alpha_{S2} < 0^\circ$ and $|\alpha_{D2}| < |\alpha_{S2}|$, the rotor speeds of drone D2 must satisfy the conditions $\omega_{M1} = \omega_{M3}$ and $\omega_{M2} > \omega_{M4}$. This is the only possibility to generate a counter reaction torque to the constraint torque applied by the cable. For the

given trajectory, only small differences in rotor speed of about $|\omega_p| \approx \omega_{M0} \pm 15$ rad/s from the hovering speed are necessary. The hovering speed can be derived for the stationary case from Equation (16), which simplifies to

$$\omega_{M0} = \sqrt{\frac{(m_D + \frac{m_P}{3})g}{4c_T}}. \quad (49)$$

For the 1000th masonry unit the hovering speed is $\omega_{M0} \approx 308.47$ rad/s. The curves of both the cable and drone angles as well as the rotor speeds have the desired smooth curve. In total, the optimization result of scenario 1 shows that the path planning framework works as expected. This means that a penalty-free solution for a complex maneuver was found.

Second scenario: The second scenario is a comparison between the drone system already presented and a cable robot. The cable robot mainly consists of a base frame to which pulleys are attached. The cables enter the actual workspace via pulleys. The platform is moved by changing the length or tension, respectively, of each of the n_c cables in a coordinated manner [66]. For the drone system, the movement of the platform results from the movement of each individual drone for constant cable lengths. The masonry units are fed to the construction area via an automated mortar system. Since the mortar system is also responsible for applying the mortar in addition to feeding the masonry units, it is also used for the drone system. For a comparison of both robotic systems, the trajectories for all 1720 masonry units of the first floor are optimized and compared in terms of time required and energy used. Due to the natural boundaries drawn by the workspace of the cable robot frame, the danger of a personal injury is little. Since it can be assumed that there are no obstacles in the workspace other than the masonry units already placed, the masonry units are transferred from the start to the target position using straight trajectories. As this simplifies numerous operations, and to maintain comparability, this trajectory shape is also used for the drone system. Accordingly, for the cable robot, the only optimization parameter is the transition time, which is why a gradient-based optimization method is suitable. For this purpose, the optimizer `fmincon` of MATLAB® is used with its default settings [67]. The parameters for the cost function are listed in Table 5.

Table 5. Optimization parameters for the cable robot.

Costfunction	Cost	Weight	Penalty	Limits
Time	1	$1 \frac{1}{s}$	-	-
Cable force	1	0	10,000	150 ... 4500 N
Cable velocity limits	1	0	1000	$6 \frac{m}{s}$
ObEE	1	0	1000	$d_{min} = 0.1$ m
CaCa	1	0	1000	-
CaEE	1	0	1000	$d_{min} = 0.1$ m
CaOb	1	0	1000	$d_{min} = 0.1$ m
Position end effector	1	0	1000	$r_{P,z} = 0 \dots 4.5$ m
Velocity end effector	1	0	1000	$ v_{max} = 3 \frac{m}{s}$
Acceleration end effector	1	0	1000	$ a_{max} = 5 \frac{m}{s^2}$
Motor torque	1	0	1000	$\tau_t = -10 \dots 110$ Nm
Power	1	0	1000	$P_{max} = 32,000$ W

Although a lower cable force limit of 150 N seems to be excessive, it is well chosen from practical experience (and thus also include a practically identified safety margin for control purposes). The drone system has the same characteristic as in the first scenario. In addition to the actual transition time t_t , the spreading factor λ_s is also optimized. In this case, as in

scenario 1, a particle swarm optimization is preferred. The parameters of the cost functions are the same as in Table 3, with the exception that $W_\chi = 0$ for $\chi \in \{ObD, CaOb\}$.

Figures 7 and 8 show the optimization results for trajectories 4 to 10.

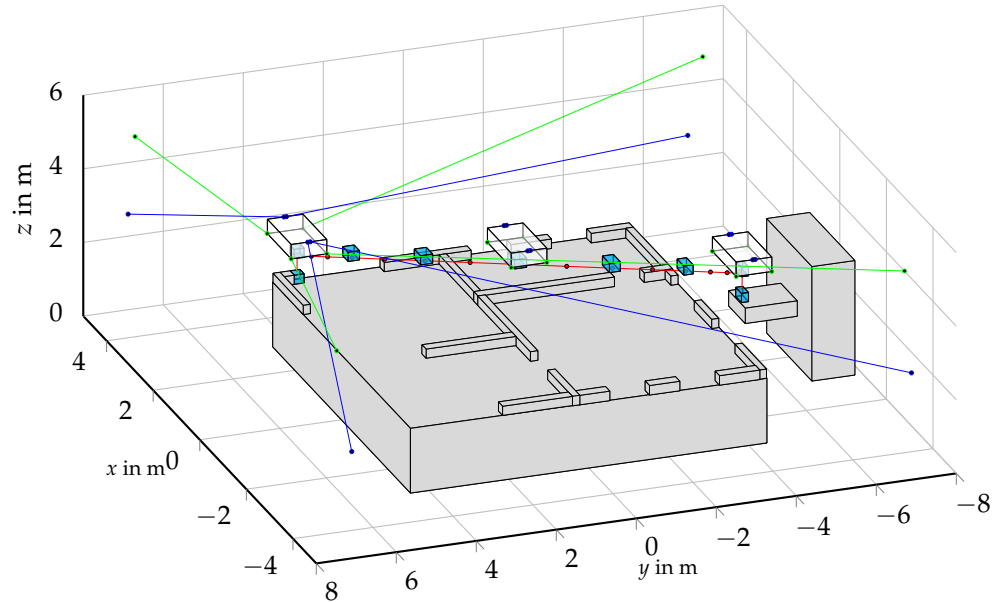


Figure 7. Optimization result for the 1000th masonry unit for scenario 2, using a cable robot.

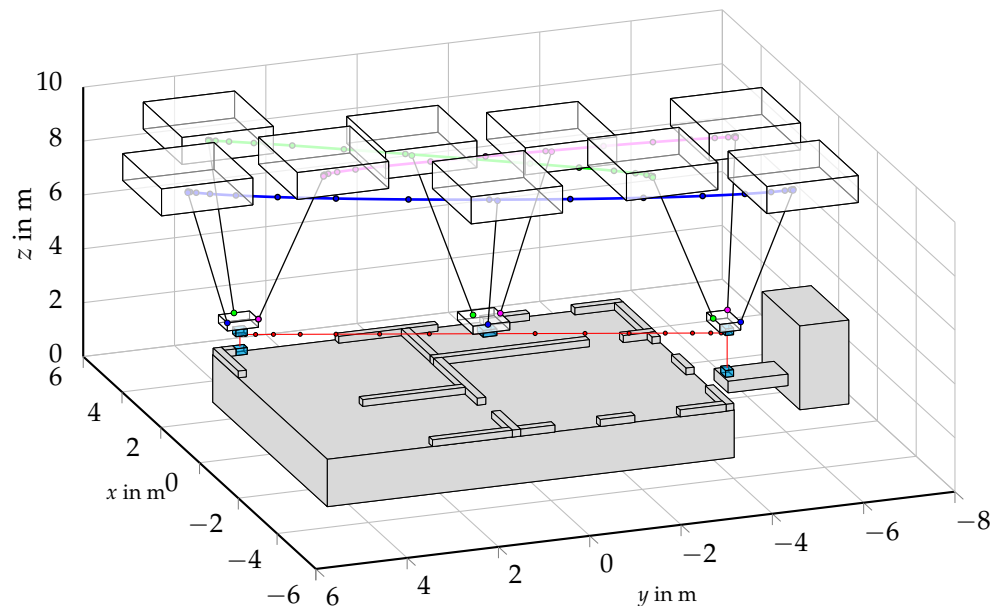


Figure 8. Optimization result for the 1000th masonry unit for scenario 2, using a drone system.

For both systems, the optimization results in the desired straight trajectories. Based on the AABBs for the end effector and the masonry unit, the result shows the different rotation methods for the masonry unit around the platform's z_P -axis. For the cable robot, it can be seen that the end effector maintains its orientation throughout the transportation while the masonry unit rotates. For the drone system, however, the rotation of the masonry unit is the same as that of the end effector to which the masonry unit is attached. For both systems, the expected change in the size of the AABBs can be seen during the rotation. Furthermore it is obvious that the cables of the cable robot are blocking relevant volumes within workspace during the motion. This increases the risk of cable collisions, to be avoided by the optimizer. For the scenario at hand, the lower pulleys have already reached a certain height. Note that

the pulleys are lifted with a rising building between two masonry units of different layers. Moreover, the transfer position of the mortar system is lifted accordingly. In contrast, the cables of the drone system are always above the end effector. Hence, cable collisions with an obstacle are almost impossible if a certain laying plan is maintained.

The resulting cable forces are presented for the cable robot in Figure 9. Analogously, the rotor speeds for the drone system are shown in Figure 10d–f.

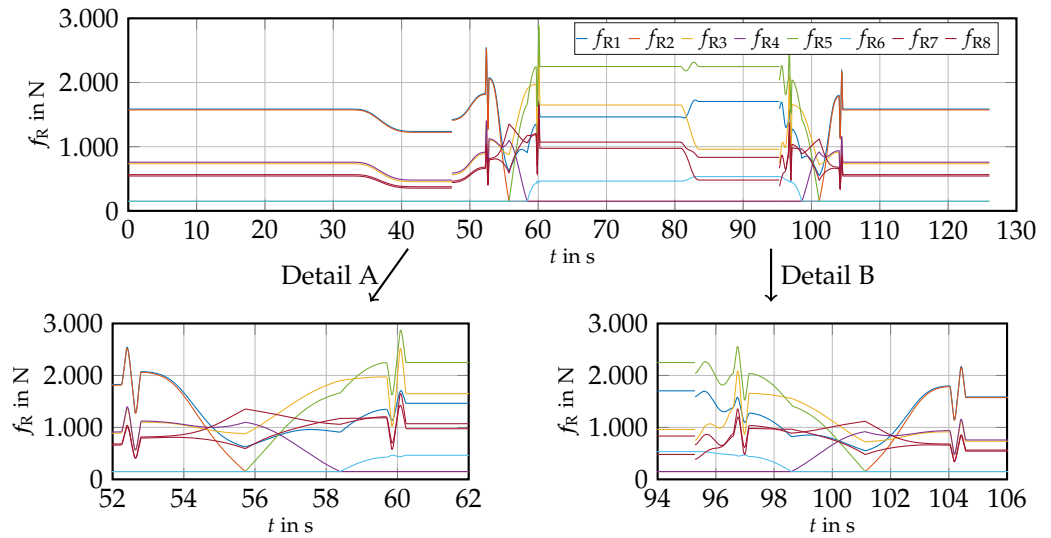


Figure 9. Calculated cable forces of the cable robot for the 1000th masonry unit.

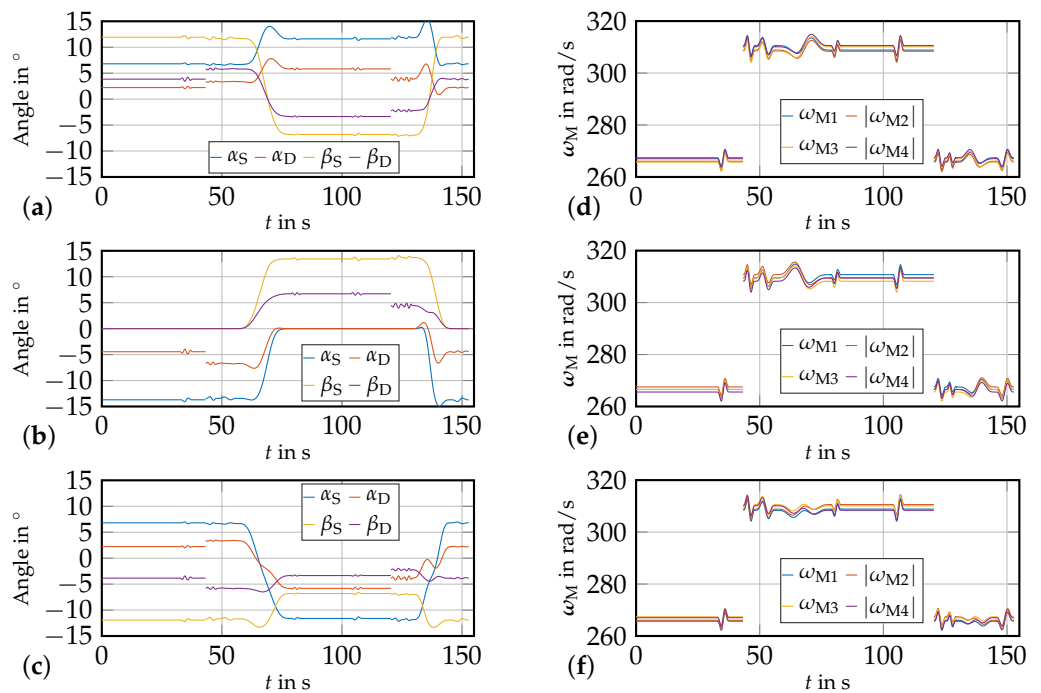


Figure 10. Resulting drone and cable angles in (a–c) and rotor speeds in (d–f) for the 1000th masonry unit using the drone system.

For the cable robot, the highest calculated cable forces are below $f_{R,max}$. This means that the optimizer was not able to accelerate the trajectory any further due to the defined penalties, although the cable robot still has cable force reserves. For both systems, the curves show a smooth course, with the exception of the transition during gripping or placing the masonry unit. The force of gravity of the platform varies in dependence of the actual masonry unit mass. For the cable robot this varies between $F_{G,P} = 1050$ N and

$F_{G,P} = 1508\text{ N}$ and for the drone system between $F_{G,D} = 301\text{ N}$ and $F_{G,D} = 723\text{ N}$. Due to the movement of the drones along with the platform, the cable forces are only slightly more than one-third of $F_{G,D}$ each. In the case of the cable robot, the cables often have a unsuitable flat angle at the platform due to the cable pulleys fixed at the frame. As a result, the net cable forces of the cable robot to carry the platform tend to be in the lower kN range and are thus many times larger than the cable forces of the drone system. Another reason for the high cable forces is that the lower cables additionally pull the platform downwards. Therefore, it follows that the cables of the drone system are subjected to much less stress. On the other hand, however, each drone of the drone system has to carry its own weight on top of the platform's weight. Depending on the platform's mass, this results in the hovering speeds $|\omega_{M0}| \approx 308.18 \dots 362.59\text{ rad/s}$ which are, compared to the theoretical speed for applying the force of gravity, slightly increased. The deviation depends on the spreading factor λ_S . Using the lift coefficient, the theoretically required force per drone of $F_A \approx 275.16\text{ N}$ to $F_A \approx 380.90\text{ N}$ can be calculated. The significantly better ratio of the force needed to carry the platform is solely due to the cable angles at the platform.

Despite the significantly larger necessary cable forces, the cable robot is more efficient due to its drive concept. With the given quantities of the cable robot, the resulting force per watt of applied power can be calculated for the stationary case. Using Equations (26)–(29) leads to the relation

$$\eta_{\text{cable}} = \frac{[N]}{[W]} = \frac{2(i_G \eta_{Gk_t})^2 \eta_{\text{ESC}}}{3r_W^2 R_P f_R} \approx \frac{2(10 \cdot 0,8941 \cdot 1.4)^2 \cdot 0.95}{3 \cdot 0.135^2 \cdot 0.12 f_R} = \frac{45374.5}{f_R} \quad (50)$$

From the manufacturer's data sheet for the drive unit of the drone system the same relationship can be derived with

$$\eta_{\text{drone}} = \frac{[N]}{[W]} = \frac{4\sqrt{c_T^3}}{2c_W \sqrt{F_A}} \approx \frac{1.243}{\sqrt{F_A}} \quad (51)$$

Common to both systems is that the ratio decreases with an increasing force. A comparison of the ratios η_{cable} and η_{drone} is done for the initial position of the 1000th masonry unit, which can be seen in Figures 9 and 10. For the cable force $f_{R1} = 1586\text{ N}$, one can calculate $\eta_{\text{cable}} = 28.61\text{ N/W}$, which is over 329 times more effective than the ratio of $\eta_{\text{drone}} = 0.087\text{ N/W}$ for the total thrust of $F_A = 205\text{ N}$ of the drone D2. Another advantage of the cable robot is its ability to recuperate energy when lowering or decelerating the masonry unit. Recuperation of stored kinetic or potential energy is currently not realistic in the case of the drone system.

Integral comparison and conclusion: From the optimization results, it follows that the overall construction time for all 1720 masonry units is $t_C = 58.85\text{ h}$ for the cable robot and $t_D = 83.09\text{ h}$ in the case of the drone system. Hence, the drone system requires approximately $\Delta t = 24.4\text{ h}$ more time, which corresponds to a value of over 41 %. With a total of 25.48 h, a lot of time is needed for the measurement processes during trajectories 1 and 8. Therefore, the actual time for moving the platform from A to B is comparatively small compared to the waiting and measurement time. Especially for the drone system, this has a very negative effect due to the high energy demand. The drone system with $E_D = 602.15\text{ kWh}$ requires more than 44 times as much energy as the cable robot with $E_C = 13.68\text{ kWh}$. This fact is mainly due to the bad ratio of applied power to the resulting lift force. In practice, this results in 186 battery changes for each drone during construction. Assuming that two minutes are needed to change the batteries of all three drones, this task would add another $t_B = 6.2\text{ h}$ to the construction time. Looking at the recuperated energy of the cable robot, the amount of energy over all masonry units is $E_{\text{rec}} = 7.94\text{ kWh}$, which is about 36.72 %. Accordingly, the recuperation capability contributes significantly to the low energy consumption of the cable robot.

Another disadvantage of the drone system is its significantly more complex positioning. While the positioning of the end effector in the case of the cable robot can be directly

determined with a high accuracy via the cable lengths, complex methods must be applied to determine the position of the drone system. Conceivable options are, for instance, the usage of satellite navigation with the application of real-time kinematic positioning or an optical tracking system. In addition, the platform with only three cables is statically indeterminate in the case of the drone system, which is why it can always be stimulated to make undesired movements by disturbances such as wind. In contrast, the platform of the cable robot with eight cables is even statically overdetermined. The drone system, however, has its advantages when it comes to flexibility, beginning with its much faster setup time. Moreover the drone system has great advantages whenever spatially flexible transports over long distances are required, as in the application of scenario 1. Here, the cable robot is not an alternative. The same applies to very large buildings. Here, the frame of the cable robot, the mortar system and the pulleys would have to be constantly readjusted. Due to the limited workspace and the unsuitable cable angles, it is even important for the cable robot that the masonry is centered in the construction space of the cable robot (which might have a climbing frame to support e.g., high rise buildings). In conclusion, it can be said that both systems are suitable for transporting heavy loads, whereas the cable robot might as well transport masonry units significantly heavier than in this scenario. In conclusion, due to the discussed differences in their characteristics, their fields of application also differ widely. While the cable robot is predestined for the movement of very heavy loads within a defined and limited workspace with a very high accuracy and/or over a long time period, a drone system has its advantages above all when the transportation has to take place over longer distances for a limited period of time.

5. Conclusions and Outlook

This paper dealt with the investigation of two different robotic systems for automated masonry construction. The comparison is based on an object-oriented programmed framework for model-based dynamical path planning. The main advantage of the object-oriented approach is the easy adaption of the framework to different scenarios, new robots or cost functions that have to be taken into account. Overall, the software tool has proven to be an efficient tool for the investigation of robots which transport a common load using cables.

The optimization result greatly depends on the definition of the cost functions and penalty terms. While the penalty terms ensure that physical boundaries are met and collisions are avoided, the quality of a penalty-free trajectory is rated by the total scalar value of the cost function itself. For the consideration of the physical boundaries like cable forces, motor limits or energy consumption, the inverse model of each robotic system was derived in detail. Additionally, the modeling for cable–cable collisions as well as cable–obstacle collisions and robot–obstacle collisions were presented. The optimization of a first scenario demonstrated the full functionality of the optimizer for transporting a payload in a complex environment using a drone system.

In a second scenario, the construction of the first floor of a building was compared for both a cable robot as well as a drone system by optimizing the trajectories of all 1720 masonry units. The result shows, on the one hand, that both robotic systems are able to transport heavy loads, but on the other hand, that they have also great differences in their characteristics. The major differences are the energy consumption and the overall construction time. These lead to different fields of applications for the two systems.

In the future, the resulting optimization data for the drone system has to be validated using a dynamic physical simulation model, which contains less simplifications. This implies, for example, that the drones cable attachment points may not coincide with their center of gravity. Moreover, the acting Coriolis and centrifugal forces of rotating parts like the propellers should be considered. In case the deviations between the rotor speed determined by the optimizer and the resulting propeller speeds from the accurate simulation model are too big, the inverse model has to be investigated for the cause. In a further step, a prototype of the drone system has to be set up. In the case of the cable robot, the next step is to validate the calculated data on a full erected prototype. This includes

mainly the cable forces and the energy consumption. To speed up the overall construction time, one future investigation will deal with the improvement of the positioning process of the end effector while gripping or placing the masonry units. This should be reached by optimizing the measurement routine of the laser sensors, in order to speed up the detection of the masonry unit's pose, as well as the detection of the erected building structure.

Author Contributions: Conceptualization, L.H., R.B. and T.B.; methodology, T.B. and R.B.; software, R.B., M.L., L.H., T.B. and S.M.; validation, L.H. and R.B.; investigation, R.B., L.H. and M.L.; resources, L.H. and T.B.; data curation, R.B. and L.H.; writing—original draft preparation, L.H., R.B. and T.B.; visualization, L.H. and T.B.; supervision, T.B.; project administration, T.B.; funding acquisition, T.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Federal Ministry of Economic Affairs and Energy (BMWi) via the Arbeitsgemeinschaft industrieller Forschungsvereinigungen “Otto von Guericke” e.V. (AiF) within the program Industrial Collective Research (IGF), funding no. 20061 BG “Entwicklung von Seilrobotern für die Erstellung von Kalksandstein-Mauerwerk auf der Baustelle”, based on a resolution of the German Bundestag. It was also funded by the Ministry of Regional Identity, Communities and Local Government, Building and Gender Equality of the Land of North Rhine-Westphalia as part of the funding “Langfristige experimentelle Untersuchung und Demonstration von automatisiertem Mauern und 3D-Druck mit Seilrobotern” and by the Ministry of Regional Identity, Communities and Local Government, Building and Digitalization of the Land of North Rhine-Westphalia within the project “Auf dem Weg zur digitalen Bauausführung: Automatisierung des Rohbaus mit Seilroboter-Technik”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BIM	Building Information Modeling
CDPR	Cable-Driven Parallel Robot
AABB	Axis-Aligned Bounding Box
OBB	Oriented Bounding Box
DOF	Degree-Of-Freedom
EU	European Union

References

1. Borrmann, A.; König, M.; Koch, C.; Beetz, J. *Building Information Modeling*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2015.
2. Templin, T.; Popielarczyk, D. (Eds.) *Proceedings of the Creative Construction e-Conference 2020*; Budapest University of Technology and Economics: Budapest, Hungary, 2020.
3. Vähä, P.; Heikkilä, T.; Kilpeläinen, P.; Järviluoma, M.; Gambao, E. Extending automation of building construction—Survey on potential sensor technologies and robotic applications. *Autom. Constr.* **2013**, *36*, 168–178. [[CrossRef](#)]
4. Brehm, E. Robots for masonry construction—Status quo and thoughts for the German market. *Mauerwerk* **2019**, *23*, 87–94. [[CrossRef](#)]
5. Bruckmann, T.; Mattern, H.; Spengler, A.; Reichert, C.; Malkwitz, A.; König, M. Automated Construction of Masonry Buildings using Cable-Driven Parallel Robots. In *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC 2016)*, Auburn, AL, USA, 18–21 July 2016. [[CrossRef](#)]
6. Wu, Y.; Cheng, H.; Fingrut, A.; Crolla, K.; Yam, Y.; Lau, D. CU-brick cable-driven robot for automated construction of complex brick structures: From simulation to hardware realisation. In *Proceedings of the 2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, Brisbane, Australia, 16–19 May 2018; pp. 166–173. [[CrossRef](#)]
7. Bruckmann, T.; Spengler, A.J.; Karl, C.K.; Reichert, C.; König, M. Process Analysis of Cable-Driven Parallel Robots for Automated Construction. In *Mechatronics for Cultural Heritage and Civil Engineering. Intelligent Systems, Control and Automation: Science and Engineering*; Springer: Cham, Switzerland, 2018; pp. 63–83. [[CrossRef](#)]

8. Heidel, R.; Lemmen, P.; Boumann, R.; Bruckmann, T. Auslegung und Inbetriebnahme eines Seilroboters zum automatisierten Mauern von Gebäudewänden. In Proceedings of the Fachtagung: VDI Mechatronik, Darmstadt, Germany, 23–24 March 2022; pp. 126–132. [\[CrossRef\]](#)
9. Roske, T.; Eden, W.; Schäfers, M.; Lemmen, P.; Heidel, R.; Boumann, R.; Spengler, A.J.; Bruckmann, T.; Lipowsky, J.; Palzer, U. Automated Construction of Calcium Silicate Masonry by Cable Robots. *AAC Worldw.* **2021**, *3*, 34–40.
10. Bosscher, P.; Williams, R.L.; Bryson, L.; Castro-Lacouture, D. Cable-Suspended Robotic Contour Crafting System. *Autom. Constr.* **2007**, *17*, 45–55. [\[CrossRef\]](#)
11. Izard, J.B.; Dubor, A.; Hervé, P.E.; Cabay, E.; Culla, D.; Rodriguez, M.; Barrado, M. Large-Scale 3D printing with Cable-Driven Parallel Robots. *Constr. Robot.* **2017**, *1*, 69–76. [\[CrossRef\]](#)
12. Izard, J.B.; Dubor, A.; Hervé, P.E.; Cabay, E.; Culla, D.; Rodriguez, M.; Barrado, M. On the Improvements of a Cable-Driven Parallel Robot for Achieving Additive Manufacturing for Construction. In *Mechanisms and Machine Science*; Springer International Publishing AG: Cham, Switzerland, 2018; Volume 53, pp. 353–363. [\[CrossRef\]](#)
13. Tenorio Ríos, J.A.; Sotorriño Ortega, G. Possibilities and Requirements for Works and Products on 3D Printing Technology. Digital Construction Week, London, UK, 2019. Available online: <http://hdl.handle.net/10261/210926> (accessed on 7 September 2022).
14. Papacharalampopoulos, A.; Bikas, H.; Foteinopoulos, P.; Stavropoulos, P. A Path Planning Optimization Framework for Concrete Based Additive Manufacturing Processes. *Procedia Manuf.* **2020**, *51*, 649–654. In Proceedings of the 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021), Athens, Greece, 15–18 June 2021. [\[CrossRef\]](#)
15. Iturralde, K.; Feucht, M.; Hu, R.; Pan, W.; Schlandt, M.; Linner, T.; Bock, T.; Izard, J.B.; Eskudero, I.; Rodriguez, M.; et al. A Cable Driven Parallel Robot with a Modular End Effector for the Installation of Curtain Wall Modules. In Proceedings of the 37th International Symposium on Automation and Robotics in Construction and Mining (ISARC), Kitakyushu, Japan, 27–28 October 2020; Tateyama, K., Ishii, K., Fumihiko, I., Eds.; The International Association for Automation and Robotics in Construction (IAARC): Tokyo, Japan, 2020; pp. 1472–1479. [\[CrossRef\]](#)
16. Bright, L.Z.; Handley, M.; Chien, I.; Curi, S.; Brownworth, L.A.; D’hers, S.; Bernier, U.R.; Gurman, P.; Elman, N.M. Analytical models integrated with satellite images for optimized pest management. *Precis. Agric.* **2016**, *17*, 628–636. [\[CrossRef\]](#)
17. Ashour, R.; Taha, T.; Mohamed, F.; Hableel, E.; Kheil, Y.A.; Elsalamouny, M.; Kadadha, M.; Rangan, K.; Dias, J.; Seneviratne, L.; Cai, G. Site inspection drone: A solution for inspecting and regulating construction sites. In Proceedings of the 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), Abu Dhabi, United Arab Emirates, 16–19 October 2016; pp. 1–4. [\[CrossRef\]](#)
18. Rizo-Maestre, C.; González-Avilés, Á.; Galiano-Garrigós, A.; Andújar-Montoya, M.D.; Puchol-García, J.A. UAV + BIM: Incorporation of Photogrammetric Techniques in Architectural Projects with Building Information Modeling Versus Classical Work Processes. *Remote. Sens.* **2020**, *12*, 2329. [\[CrossRef\]](#)
19. Suarez Fernandez, R.A.; Dominguez, S.; Campoy, P. L1 adaptive control for Wind gust rejection in quad-rotor UAV wind turbine inspection. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1840–1849. [\[CrossRef\]](#)
20. Vanderhorst, H.R.; Suresh, S.; Suresh, R. Special Issue. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 416–428. [\[CrossRef\]](#)
21. Placke, T.; Kloepsch, R.; Dühnen, S.; Winter, M. Lithium ion, lithium metal, and alternative rechargeable battery technologies: The odyssey for high energy density. *J. Solid State Electrochem.* **2017**, *21*, 1939–1964. [\[CrossRef\]](#)
22. Olivares, V.; Cordova, F.; Sepúlveda, J.M.; Derpich, I. Modeling Internal Logistics by Using Drones on the Stage of Assembly of Products. *Procedia Comput. Sci.* **2015**, *55*, 1240–1249. [\[CrossRef\]](#)
23. Apeland, J.; Pavlou, D.; Hemmingsen, T. State-of-Technology and Barriers for Adoption of Fuel Cell Powered Multirotor Drones. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 1359–1367. [\[CrossRef\]](#)
24. Liang, X.; Fang, Y.; Sun, N.; Lin, H.; Zhao, X. Adaptive Nonlinear Hierarchical Control for a Rotorcraft Transporting a Cable-Suspended Payload. In *IEEE Transactions on Systems, Man, and Cybernetics. Systems*; IEEE: New York, NY, USA, 2020; pp. 1–12. [\[CrossRef\]](#)
25. Augugliaro, F.; Lupashin, S.; Hamer, M.; Male, C.; Hehn, M.; Mueller, M.; Willmann, J.S.; Gramazio, F.; Kohler, M.; D’Andrea, R. The Flight Assembled Architecture installation: Cooperative construction with flying machines. *IEEE Control Syst.* **2014**, *34*, 46–64. [\[CrossRef\]](#)
26. Naldi, R.; Gentili, L.; Marconi, L. Modeling and control of the interaction between flying robots and the environment. *IFAC Proc. Vol.* **2010**, *43*, 975–980. [\[CrossRef\]](#)
27. Pounds, P.E.I.; Bersak, D.R.; Dollar, A.M. Grasping from the air: Hovering capture and load stability. In Proceedings of the IEEE International Conference 2011, Shanghai, China, 9–13 May 2011; Volume 2011, pp. 2491–2498. [\[CrossRef\]](#)
28. Wang, Y.; Cai, H.; Zhang, J.; Li, X. Disturbance Attenuation Predictive Optimal Control for Quad-Rotor Transporting Unknown Varying Payload. *IEEE Access* **2020**, *8*, 44671–44686. [\[CrossRef\]](#)
29. Ryll, M.; Muscio, G.; Pierri, F.; Cataldi, E.; Antonelli, G.; Caccavale, F.; Franchi, A. 6D physical interaction with a fully actuated aerial robot. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5190–5195. [\[CrossRef\]](#)

30. Muscio, G.; Pierri, F.; Trujillo, M.A.; Cataldi, E.; Giglio, G.; Antonelli, G.; Caccavale, F.; Viguria, A.; Chiaverini, S.; Ollero, A. Experiments on coordinated motion of aerial robotic manipulators. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1224–1229. [CrossRef]
31. Tognon, M.; Yuksel, B.; Buondonno, G.; Franchi, A. Dynamic decentralized control for protocentric aerial manipulators. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 6375–6380. [CrossRef]
32. Alothman, Y.; Gu, D. Using Constrained Model Predictive Control to Control Two Quadrotors Transporting a Cable-Suspended Payload. In Proceedings of the 2018 13th World Congress on Intelligent Control and Automation (WCICA), Changsha, China, 4–8 July 2018; pp. 228–233. [CrossRef]
33. Gassner, M.; Cieslewski, T.; Scaramuzza, D. Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5196–5202. [CrossRef]
34. Heyden, T.; Maier, T.; Woernle, C., Trajectory Tracking Control for a Cable Suspension Manipulator. In *Advances in Robot Kinematics: Theory and Applications*; Lenarčič, J., Thomas, F., Eds.; Springer: Dordrecht, The Netherlands, 2002; pp. 125–134. [CrossRef]
35. Michael, N.; Fink, J.; Kumar, V. Cooperative manipulation and transportation with aerial robots. *Auton. Robot.* **2011**, *30*, 73–86. [CrossRef]
36. Manubens, M.; Devaurs, D.; Ros, L.; Cortés, J. Motion Planning for 6-D Manipulation with Aerial Towed-cable Systems. In Proceedings of the Robotics: Science and Systems IX. Robotics: Science and Systems Foundation, Berlin, Germany, 24–28 June 2013. [CrossRef]
37. Mohammadi, K.; Sirouspour, S.; Grivani, A. Control of Multiple Quad-Copters With a Cable-Suspended Payload Subject to Disturbances. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1709–1718. [CrossRef]
38. Masone, C.; Bulthoff, H.H.; Stegagno, P. Cooperative transportation of a payload using quadrotors: A reconfigurable cable-driven parallel robot. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1623–1630. [CrossRef]
39. Hashemi, D.; Heidari, H. Trajectory Planning of Quadrotor UAV with Maximum Payload and Minimum Oscillation of Suspended Load Using Optimal Control. *J. Intell. Robot. Syst.* **2020**, *100*, 1369–1381. [CrossRef]
40. Qian, L.; Liu, H.H. Path-Following Control of A Quadrotor UAV With A Cable-Suspended Payload Under Wind Disturbances. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2021–2029. [CrossRef]
41. Guo, M.; Gu, D.; Zha, W.; Zhu, X.; Su, Y. Controlling a Quadrotor Carrying a Cable-Suspended Load to Pass Through a Window. *J. Intell. Robot. Syst.* **2020**, *98*, 387–401. [CrossRef]
42. Foehn, P.; Falanga, D.; Kuppaswamy, N.; Tedrake, R.; Scaramuzza, D. Fast Trajectory Optimization for Agile Quadrotor Maneuvers with a Cable-Suspended Payload. In Proceedings of the Robotics: Science and Systems XIII. Robotics: Science and Systems Foundation, Cambridge, MA, USA, 12–16 July 2017. [CrossRef]
43. Cruz, P.; Fierro, R. Autonomous lift of a cable-suspended load by an unmanned aerial robot. In Proceedings of the 2014 IEEE Conference on Control Applications (CCA), Juan Les Antibes, France, 8–10 October 2014; pp. 802–807. [CrossRef]
44. Cruz, P.J.; Oishi, M.; Fierro, R. Lift of a cable-suspended load by a quadrotor: A hybrid system approach. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 1887–1892. [CrossRef]
45. The MathWorks, Inc. Global Optimization Toolbox. Available online: <https://www.mathworks.com/help/gads/> (accessed on 7 September 2022).
46. Benson, H.P., Multi-Objective Optimization: Pareto Optimal Solutions, Properties. In *Encyclopedia of Optimization*; Floudas, C.A., Pardalos, P.M., Eds.; Springer: Boston, MA, USA, 2009; pp. 2478–2481. [CrossRef]
47. Mattern, H.; Bruckmann, T.; Spengler, A.J.; König, M. Simulation of Automated Construction using Wire Robots. In *Proceedings of the 2016 Winter Simulation Conference*; Roeder, T.M., Frazier, P.I., Szechtman, R., Zhou, E., Huschka, T., Chick, S.E., Eds.; IEEE Press: Arlington, VA, USA, 2016; pp. 3302–3313. [CrossRef]
48. Bruckmann, T.; Boumann, R. Simulation and optimization of automated masonry construction using cable robots. *Adv. Eng. Inform.* **2021**, *50*, 101388. [CrossRef]
49. Pott, A. *Cable-Driven Parallel Robots: Theory and Application*; Springer: Cham, Switzerland, 2018. [CrossRef]
50. Mahony, R.; Kumar, V.; Corke, P. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–32. [CrossRef]
51. Herrmann, L.; Bröcker, M. Flatness based control of a ball in tube system. *IFAC-Papers OnLine* **2015**, *48*, 790–795. [CrossRef]
52. Pillay, P.; Krishnan, R. Application characteristics of permanent magnet synchronous and brushless DC motors for servo drives. *IEEE Trans. Ind. Appl.* **1991**, *27*, 986–996. [CrossRef]
53. Bruckmann, T.; Reichert, C.; Ji, H. Energy Consumption Reduction of a Cable-Driven Storage and Retrieval System. In *Advances in Robot Kinematics 2018*; Lenarčič, J., Parenti-Castelli, V., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 383–391. [CrossRef]

54. Reichert, C.; Glogowski, P.; Bruckmann, T. Dynamische Rekonfiguration eines seilbasierten Manipulators zur Verbesserung der mechanischen Steifigkeit. In *Fachtagung Mechatronik 2015: Dortmund (12–13 March 2015), Proceedings*; Bertram, T., Corves, B., Janschek, K., Eds.; Institute of Mechanism Theory, Machine Dynamics and Robotics: Aachen, Germany, 2015; pp. 91–96. [[CrossRef](#)]
55. Ericson, C. *Real-Time Collision Detection*; The Morgan Kaufmann Series in Interactive 3D Technology; Morgan Kaufmann: San Francisco, CA, USA, 2005; p. iv. [[CrossRef](#)]
56. Ericson, C. *Real-Time Collision Detection*; CRC Press: Boca Raton, FL, USA, 2004.
57. Nguyen, D.Q.; Gouttefarde, M. On the Improvement of Cable Collision Detection Algorithms; Cable-Driven Parallel Robots; Pott, A., Bruckmann, T., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 29–40.
58. Merlet, J.P. Analysis of the Influence of Wires Interference on the Workspace of Wire Robots. In *On Advances in Robot Kinematics*; Lenarčič, J., Galletti, C., Eds.; Springer: Dordrecht, The Netherlands, 2004; pp. 211–218.
59. Merlet, J.P.; Daney, D. Legs interference checking of parallel robots over a given workspace or trajectory. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 757–762. [[CrossRef](#)]
60. Perreault, S.; Cardou, P.; Gosselin, C.M.; Otis, M.J.D. Geometric Determination of the Interference-Free Constant-Orientation Workspace of Parallel Cable-Driven Mechanisms. *J. Mech. Robot.* **2010**, *2*, 031016. [[CrossRef](#)]
61. Blanchet, L.; Merlet, J.P. Interference detection for cable-driven parallel robots (CDPRs). In Proceedings of the 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besancon, France, 8–11 July 2014; pp. 1413–1418. [[CrossRef](#)]
62. Rich, H. *Kalksandstein: Die Maurerfibel*, 8th ed.; Verlag Bau+Technik GmbH: Düsseldorf, Germany, 2014.
63. *Deutsches Institut für Normung e.V. DIN V 106:2005-10: Calcium Silicate Units with Specific Properties*; Beuth-Verlag: Berlin, Germany, 2005. [[CrossRef](#)]
64. Heyden, T.; Woernle, C. Flatness-Based Trajectory Tracking Control of an Underconstrained Cable Suspension Manipulator. *PAMM* **2004**, *4*, 129–130. [[CrossRef](#)]
65. The MathWorks, Inc. Particle Swarm. Available online: <https://www.mathworks.com/help/gads/particle-swarm.html> (accessed on 7 September 2022).
66. Gouttefarde, M.; Lamaury, J.; Reichert, C.; Bruckmann, T. A Versatile Tension Distribution Algorithm for n -DOF Parallel Robots Driven by $n + 2$ Cables. *IEEE Trans. Robot.* **2015**, *31*, 1444–1457. [[CrossRef](#)]
67. The MathWorks, Inc. Fmincon. Available online: <https://www.mathworks.com/help/optim/ug/fmincon.html> (accessed on 7 September 2022).

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub | universitäts
bibliothek

This text is made available via DuEPublico, the institutional repository of the University of Duisburg-Essen. This version may eventually differ from another version distributed by a commercial publisher.

DOI: 10.3390/robotics11060119

URN: urn:nbn:de:hbz:465-20230421-111547-1



This work may be used under a Creative Commons Attribution 4.0 License (CC BY 4.0).