Optimization-Based Multipoint Trajectory Planning Along Straight Lines for Tower Cranes

Mark Burkhardt, Andreas Gienger, and Oliver Sawodny, *Senior Member, IEEE*

***Abstract*— On common construction sites, it is often the case that tower cranes must move heavy payloads along straight lines either due to obstacles or it is the direct route to a target location. In order to enable this motion for autonomous tower cranes, the goal of this work is an offline trajectory planning algorithm for the tower crane's payload to follow multiple straight connection lines of waypoints and compute smooth transitions at the intersection of the connection lines. The considered tower crane is a rigid tower crane with five degrees of freedom. An optimal control problem (OCP) is formulated in order to compute the trajectory for the transition between two waypoints minimizing the transition time and weighted path error. The smooth transitions between the connection lines are obtained by solving the OCP for the next connection line utilizing the current position as initial condition. The simulation results investigate the time gain in comparison to a trajectory that precisely positions the payload at each waypoint. The experimental validation with a real large-scale tower crane verifies that the crane is able to time-efficiently pass through an obstacle course.**

***Index Terms*— Electromechanical systems, optimal control, predictive control for nonlinear systems, trajectory planning.**

I. Introduction

THE building industry represents the largest industry globally, but faces a challenge given by the growing need of new housing, workplaces and infrastructure due to the rapid urbanization and population growth. It is all the more striking that the degree of automation of many construction machines is low in comparison to, e.g., machines utilized in the manufacturing sector. Tower cranes are crucial machines that are responsible for the overall construction progress. Tower cranes transport heavy payloads from a pick up location to a target location. The automation of such transportation tasks offers potential for increased flexibility and productivity.

One of the main control issues is to find a collision-free trajectory of the payload's coordinates path to the desired target position. Moving a load along straight sections is desirable in practice, as construction sites and constructed buildings often only allow transports along narrow corridors. In some cases it is desired to position the payload at the corner points of the straight section, whereas in other cases it is desired to continue directly without stopping. We consider the trajectory generation problem that waypoints (representing the end points of the straight sections) have been identified and the payload is supposed to follow the straight connection lines between them. The final waypoint is defined as operation point at which the payload is stopped because the payload is either mounted or dismounted. First, the path of the corresponding trajectory needs to maintain a small discrepancy between the straight connection lines of the waypoints while the trajectory also needs to smooth the connections' intersections in order to avoid a braking process for the payload such that the transition time is not unnecessarily prolonged. We assume that the smoothed path is required to maintain a user-specified distance to the intersection of the connection lines. Second, maneuvering the transported payload yields load sway motions that require time to manually damp and present a danger. Thus, the tower crane's sway dynamics and actuator constraints need to be considered throughout the trajectory generation. The tower crane shown in Fig. 1 serves as the test bench for the experimental validation of the presented results.

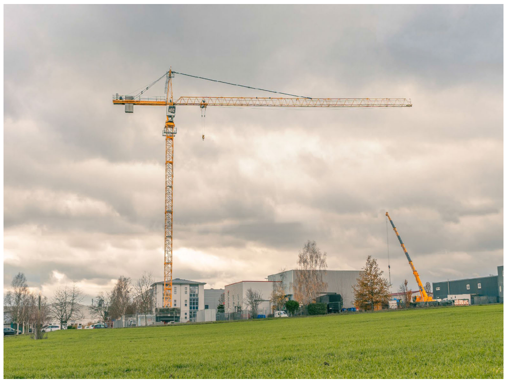
Fig. 1. Experimental tower crane Liebherr 154 EC-H Litronic 6, which is utilized on common construction sites. ©IntCDC, University of Stuttgart.

Cranes are classified as either overhead cranes or rotary cranes. The majority of the literature focuses on the trajectory planning for overhead cranes [1], [2], [3], or [4]. The optimization problem proposed in [2] computes a trajectory for moving the payload to a specified position with minimal transition time in 2-D considering acceleration and sway angle constraints. Similar to the solution presented in this brief, the optimization variable is the Cartesian payload position and the constraints are expressed with respect to the Cartesian payload position. However, neither the generation of a trajectory describing the motion in proximity to a path is solved nor the planning of time efficient transitions between a sequence of waypoints is solved.

Manuscript received 3 July 2023; accepted 16 August 2023. Date of publication 25 September 2023; date of current version 29 December 2023. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2120/1—390831618. Recommended by Associate Editor S. Galeani. *(Corresponding author: Mark Burkhardt.)*

The authors are with the Institute for System Dynamics, University of Stuttgart, 70563 Stuttgart, Germany (e-mail: burkhardt@isys.uni-stuttgart.de; gienger@isys.uni-stuttgart.de; sawodny@isys.uni-stuttgart.de).

Digital Object Identifier 10.1109/TCST.2023.3308762

Thus, the method presented in [2] is a good introduction to the method presented in this brief.

The trajectory generation for tower cranes is barely addressed in [5], [6], [7], [8], and [9]. The motion along connection lines and the planning of time efficient transitions between them while satisfying constraints are not considered.

A good overview on general trajectory planning algorithms in robotics is given in [10] and [11]. A well-known method that is often employed for multipoint trajectory generation are B-splines, see e.g., [11] and [12]. Examples for a generation of smooth trajectories between multiple waypoints are given in [13], [14], and [15]. In the context of our trajectory generation issue, B-splines can be used to obtain smooth transitions between the connection lines. However, the optimal time step during which the trajectory should switch to the B-spline depicting the transition is not known. Although this approach solves the trajectory generation issue discussed in this brief, this work's algorithm is more beneficial because the beginning of the transition between connection lines is computed inherently considering the tower crane dynamics and desired requirements.

This work presents an optimization-based trajectory generation approach minimizing the path error and transition time while considering sway angle, velocity, and acceleration constraints. A smooth transition of the trajectory between the connection lines is obtained by solving the corresponding optimization problem for tracking the connection line between the next pair of waypoints. The trajectory is planned in flat coordinates. Introductory literature on flat systems is given by [16], [17], and [18]. The improvement in efficiency of solving dynamic optimization problems through a problem formulation using flat coordinates is investigated in [19].

This work has three main contributions as follows.

1) The trajectory planning problem for a payload following the straight connection line between two waypoints with a tower crane.
2) A novel trajectory planning algorithm produces time-efficient and smooth trajectories at the transition between two straight connection lines approaching each waypoint up to a user-defined minimum distance.
3) The application and experimental validation considering an obstacle course for a large-scale tower crane.

The brief is structured as follows: Section II covers the preliminaries. Section III introduces the optimal control problem (OCP) that computes the desired payload trajectory for following the connection line between two waypoints. The main result of this brief is presented in Section IV dealing with the algorithm computing a smooth transition between the connection lines. The exemplary trajectory for the transition along three waypoints is discussed in Section V, followed by a performance demonstration of the trajectory on a large-scale experimental tower crane in Section VI. The brief is closed by a summary and conclusion given in Section VII.

II. Problem Statement

This section introduces the underlying tower crane model and the trajectory design objectives.

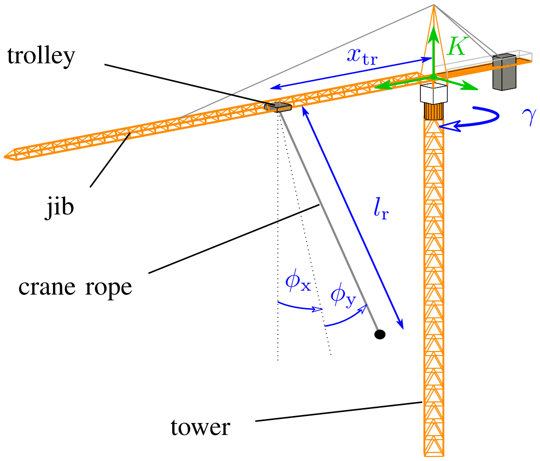

Fig. 2. Schematic drawing of the tower crane investigated throughout this work. The degrees of freedom are highlighted in blue. The inertial system is highlighted in green.

A. System Dynamics

The considered tower crane is a top-slewing tower crane as shown in Fig. 2. It consists of a horizontal jib clamped into the vertical tower. The jib and the tower are assumed to be rigid bodies. The coordinate system K serves as inertial system and is located above the tower crane cabin. The tower crane has three electrical actuators. First, the slewing drive rotates the entire jib on a slewing ring which yields the slewing angle γ. Second, the trolley drive moves along the jib, and its position in radial direction is x_{tr}. The crane hook is attached to the end of the hanging rope. The length of the crane rope l_{r} is changed by the third actuator namely the hoisting drive. The payload is modeled as a point mass, which is attached to the crane hook, whereas the hook mass is neglected due to its relatively low weight. The rope behaves like a pendulum leading to load sway during transport. The sway angles are denoted by ϕ_{x} in the tangential direction (perpendicular to the jib) and ϕ_{y} in radial direction. In total, the considered tower crane has five degrees of freedom: three actuated degrees of freedom, which are γ, x_{tr} and l_{r}, and two nonactuated degrees of freedom, which are the sway angles ϕ_{x} and ϕ_{y}. The degrees of freedom are addressed by $\mathbf{q} = [\gamma, x_{\mathrm{tr}}, l_{\mathrm{r}}, \phi_{\mathrm{x}}, \phi_{\mathrm{y}}]$. The inputs are the accelerations of the drive systems denoted by $u_\gamma = \ddot{\gamma}$, $u_{\mathrm{tr}} = \ddot{x}_{\mathrm{tr}}$, and $u_{\mathrm{r}} = \ddot{l}_{\mathrm{r}}$. The derivation of the dynamic equations for the sway angles $\ddot{\phi}_{\mathrm{x}}$ and $\ddot{\phi}_{\mathrm{y}}$ of rigid tower cranes is straightforward. The resulting equations of motion are given by (3e) and (3f) in [20]. Introducing the state vector $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}]$ and the input vector $\mathbf{u} = [u_\gamma, u_{\mathrm{tr}}, u_{\mathrm{r}}]$, it is straightforward to obtain the system in the state space representation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \tag{1}$$

Additionally, each driving system has velocity and acceleration constraints. This implies the box constraints

$$\dot{\gamma}_{\min} \le \dot{\gamma} \le \dot{\gamma}_{\max}, \quad \ddot{\gamma}_{\min} \le \ddot{\gamma} \le \ddot{\gamma}_{\max} \tag{2a}$$

$$\dot{x}_{\mathrm{tr,min}} \le \dot{x}_{\mathrm{tr}} \le \dot{x}_{\mathrm{tr,max}}, \quad \ddot{x}_{\mathrm{tr,min}} \le \ddot{x}_{\mathrm{tr}} \le \ddot{x}_{\mathrm{tr,max}} \tag{2b}$$

$$\dot{l}_{\mathrm{r,min}} \le \dot{l}_{\mathrm{r}} \le \dot{l}_{\mathrm{r,max}}, \quad \ddot{l}_{\mathrm{r,min}} \le \ddot{l}_{\mathrm{r}} \le \ddot{l}_{\mathrm{r,max}} \tag{2c}$$

for the state and input vector. For comfort and safety issues, constraints for the sway angles

$$\phi_{\mathrm{x,min}} \leq \phi_{\mathrm{x}} \leq \phi_{\mathrm{x,max}}, \quad \phi_{\mathrm{y,min}} \leq \phi_{\mathrm{y}} \leq \phi_{\mathrm{y,max}} \tag{3}$$

are also introduced. Throughout this work, the sway angle constraints are chosen to $\phi_{\mathrm{x,max}} = \phi_{\mathrm{y,max}} = -\phi_{\mathrm{x,min}} = -\phi_{\mathrm{y,min}} = 2.5°$. Therefore, the sway angles will remain small, and the rope's bending in the lateral direction is neglected.

B. Problem Definition and Requirements

Given a set of N_{wp} waypoints

$$\mathcal{W} = \left\{\mathbf{w}_1, \dots, \mathbf{w}_{N_{\mathrm{wp}}}\right\}$$

where each waypoint $\mathbf{w}_i \in \mathbb{R}^3$ describes a load position in Cartesian coordinates within the coordinate system K shown in Fig. 2 and $\mathbf{w}_{N_{\mathrm{wp}}}$ is considered as operation point. The problem considered in this brief is to generate a reference trajectory of the state vector $\mathbf{x}_{\mathrm{d}}(t)$ and a feed-forward control input $\mathbf{u}_{\mathrm{FF}}(t)$ for a Two-Degree-of-Freedom control loop describing the entire transition of waypoint $\mathbf{w}_1$ to operation point $\mathbf{w}_{N_{\mathrm{wp}}}$. The waypoints can be defined such that offsets or deformations of the tower crane's jib, tower, or rope will be compensated. For instance, a waypoint can be shifted upward in the vertical direction by the amount of bending of the tower crane's jib. The requirements on the trajectory are as follows.

1) The trajectory needs to satisfy $\mathbf{x}_{\mathrm{d}}(t) \in C^0$ and $\mathbf{u}_{\mathrm{d}} \in C^0$, i.e., it needs to be continuous.
2) The corresponding geometrical path of the payload position needs to approximate the path obtained by connecting two waypoints by a straight line.
3) The trajectory needs to satisfy the velocity and acceleration constraints for the actuated degrees of freedom (2) and the constraints of the sway angles (3) at any time.
4) During the transition to each waypoint, the geometrical path of the payload trajectory must approach each waypoint $\mathbf{w}_i$ at least at one time step t_i by a user-defined minimum distance $d_{\mathrm{wp,i}}$ serving as upper bound, i.e.,

$$\exists t_i \;:\; \left|\mathbf{r}_{\mathrm{pl,d}}(t_i) - \mathbf{w}_i\right| \leq d_{\mathrm{wp,i}} \tag{4}$$

where $\mathbf{r}_{\mathrm{pl,d}}(t)$ describes the Cartesian payload position that follows from the state trajectory $\mathbf{x}_{\mathrm{d}}(t)$.

III. Trajectory Generation for the Connection Line of Two Waypoints

This section presents an OCP utilized for generating the trajectories $\mathbf{x}_{\mathrm{d}}(t)$ and $\mathbf{u}_{\mathrm{FF}}(t)$ achieving a payload transport in proximity to the straight connection line of two waypoints $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$. To investigate the advantage of the OCP's formulation in flat coordinates over its formulation in original coordinates, a simulation study with 173 randomized transitions between waypoints has been carried out. The average computation time in flat coordinates (18 016 decision variables and 23 038 constraints for each OCP) was 67.78% (absolute values 42.87 s and 133.07 s) smaller than the average computation time in original coordinates (16 014 decision variables and 21 034 constraints for each OCP) using an Intel Core i7-8650 1.9 GHz and 2.11 Ghz processor with 16 GB RAM, Windows 10, and MATLAB 2019b. Moreover, the median computation time and average deviation of the mean were given by 19.14 s and 36.79 s using the formulation in flat coordinates compared to 143.48 s and 76.63 s using the formulation in original coordinates. In fact, employing flat coordinates yields dynamics that are integrator chains. Furthermore, by planning the Cartesian payload position directly, it is easier to include the distance to the straight connection line in the cost function of the OCP. A disadvantage is that the velocity, acceleration, and sway angle constraints need to be formulated utilizing Cartesian coordinates yielding nonlinear inequality constraints instead of box constraints. However, the easier cost function and system dynamics outweigh the nonlinear inequality constraints. Due to the computational benefits, the formulation in flat coordinates is further investigated in the following.

The reformulation of the constraints using Cartesian coordinates of the payload is given in Section III-A, whereas the OCP solving the trajectory generation for the motion along a connection line of two waypoints is presented in Section III-B.

A. Transformation of Constraints Exploiting Flatness

To include the velocity and acceleration constraints (2) and sway angle constraints (3) into the OCP, a transformation is required expressing the slewing, trolley, and hoisting velocity and acceleration with the Cartesian coordinates of the payload.

We exploit flatness for computing this transformation. A nonlinear system $\dot{\mathbf{w}} = \mathbf{f}(\mathbf{w}, \mathbf{v})$ with state vector $\mathbf{w} \in \mathbb{R}^n$ and input vector $\mathbf{v} \in \mathbb{R}^m$ is called (differentially) flat, if there are differentially independent outputs $\mathbf{h}(\mathbf{w}, \mathbf{v}, \dot{\mathbf{v}}, \dots, \mathbf{v}^{(r)}) = [h_1, \dots, h_m]$, which can be used to parameterize the state and input as

$$\mathbf{w} = \psi_{\mathbf{w}}\left(h_1, \dot{h}_1, \dots, h_1^{(\beta_1-1)}, \dots, h_m, \dot{h}_m, \dots, h_m^{(\beta_m-1)}\right) \tag{5a}$$

$$\mathbf{v} = \psi_{\mathbf{v}}\left(h_1, \dot{h}_1, \dots, h_1^{(\beta_1)}, \dots, h_m, \dot{h}_m, \dots, h_m^{(\beta_m)}\right) \tag{5b}$$

see [21]. Thereby, the derivative orders must satisfy $\sum_{i=1}^{m} \beta_i \geq n$. It is assumed that $\mathbf{f}$ is a smooth vector field. Note that $\mathbf{h}$, $\Psi_{\mathbf{w}}$, and $\Psi_{\mathbf{v}}$ are smooth functions. In this work, the flat system is given by (1) and the corresponding output equation is

$$\mathbf{r}_{\mathrm{pl}} = \begin{bmatrix} \mathrm{c}(\gamma)x_{\mathrm{tr}} - l_{\mathrm{r}}\left(\mathrm{s}(\gamma)\mathrm{s}(\phi_{\mathrm{x}}) + \mathrm{c}(\gamma)\mathrm{c}(\phi_{\mathrm{x}})\mathrm{s}(\phi_{\mathrm{y}})\right) \\ \mathrm{s}(\gamma)x_{\mathrm{tr}} + l_{\mathrm{r}}\left(\mathrm{c}(\gamma)\mathrm{s}(\phi_{\mathrm{x}}) - \mathrm{s}(\gamma)\mathrm{c}(\phi_{\mathrm{x}})\mathrm{s}(\phi_{\mathrm{y}})\right) \\ -l_{\mathrm{r}}\mathrm{c}(\phi_{\mathrm{x}})\mathrm{c}(\phi_{\mathrm{y}}) \end{bmatrix} \tag{6}$$

which describes the Cartesian position of the payload and is not unique [22]. The abbreviations $\sin(\cdot) = \mathrm{s}(\cdot)$ and $\cos(\cdot) = \mathrm{c}(\cdot)$ are employed in (6). Because the actuator velocities, accelerations, and sway angles are part of the state and input vector of the dynamics (1), the desired transformation equals to the state and input parameterization.

The state and input parameterizations for a rigid tower crane are derived in previous work [23], which is thus omitted here due to spatial constraints. The state parameterization is denoted by

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = \Psi_{\mathbf{x}}\left(\mathbf{r}_{\mathrm{pl}}, \dot{\mathbf{r}}_{\mathrm{pl}}, \ddot{\mathbf{r}}_{\mathrm{pl}}, \dddot{\mathbf{r}}_{\mathrm{pl}}\right) \tag{7}$$

and the input parameterization is denoted by

$$\mathbf{u} = \begin{bmatrix} \ddot{\gamma}\left(\mathbf{r}_{\mathrm{pl}}, \dots, \mathbf{r}_{\mathrm{pl}}^{(4)}\right) \\ \ddot{x}_{\mathrm{tr}}\left(\mathbf{r}_{\mathrm{pl}}, \dots, \mathbf{r}_{\mathrm{pl}}^{(4)}\right) \\ \ddot{l}_{\mathrm{r}}\left(\mathbf{r}_{\mathrm{pl}}, \dots, \mathbf{r}_{\mathrm{pl}}^{(4)}\right) \end{bmatrix} = \Psi_{\mathbf{u}}\left(\mathbf{r}_{\mathrm{pl}}, \dots, \mathbf{r}_{\mathrm{pl}}^{(4)}\right). \tag{8}$$

The payload position and its four derivatives in Cartesian coordinates are defined as flat coordinates

$$\mathbf{z} = \left[\mathbf{r}_{\mathrm{pl}}, \ldots, \mathbf{r}_{\mathrm{pl}}^{(4)}\right]^{\mathrm{T}} \tag{9}$$

and the expressions for each variable

$$\xi \in \left\{\gamma, \dot{\gamma}, \ddot{\gamma}, x_{\mathrm{tr}}, \dot{x}_{\mathrm{tr}}, \ddot{x}_{\mathrm{tr}}, l_{\mathrm{r}}, \dot{l}_{\mathrm{r}}, \ddot{l}_{\mathrm{r}}, \phi_{\mathrm{x}}, \dot{\phi}_{\mathrm{x}}, \phi_{\mathrm{y}}, \dot{\phi}_{\mathrm{y}}\right\}$$

with respect to the payload position and its derivatives are denoted by $\xi = \Psi_{\xi}(\mathbf{z})$.

B. Optimal Control Problem for Trajectory Generation

The discretized version of the following OCP is solved at an arbitrary time step t_0 to obtain the trajectory for a motion between two waypoints $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$:

$$\min_{t_{\mathrm{f}}, \mathbf{v}(t)} t_{\mathrm{f}} + \int_{t_0}^{t_0+t_{\mathrm{f}}} \omega_{\mathrm{dist}} d^2(\mathbf{r}_{\mathrm{pl,d}}(t), \mathbf{w}_i, \mathbf{w}_{i+1}) + \mathbf{v}^{\mathrm{T}}(t)\mathbf{R}\mathbf{v}(t)dt \tag{10a}$$

$$\text{s.t. } \dot{\mathbf{z}}_{\mathrm{d}} = \begin{bmatrix} 0 & I_3 & 0 & 0 & 0 \\ 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 0 & I_3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{z}_{\mathrm{d}} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ I_3 \end{bmatrix} \mathbf{v} \tag{10b}$$

$$\mathbf{z}_{\mathrm{d}}(t_0) = \mathbf{z}_0 \tag{10c}$$

$$\mathbf{z}_{\mathrm{d}}(t_{\mathrm{f}}) = \left[\mathbf{w}_{i+1}, 0, 0, 0, 0\right]^{\mathrm{T}} \tag{10d}$$

$$\begin{bmatrix} \phi_{\mathrm{x,min}} \\ \phi_{\mathrm{y,min}} \\ \dot{\gamma}_{\mathrm{min}} \\ \ddot{\gamma}_{\mathrm{min}} \\ \dot{x}_{\mathrm{tr,min}} \\ \ddot{x}_{\mathrm{tr,min}} \\ \dot{l}_{\mathrm{r,min}} \\ \ddot{l}_{\mathrm{r,min}} \end{bmatrix} \leq \begin{bmatrix} \Psi_{\phi_{\mathrm{x}}}(\mathbf{z}_{\mathrm{d}}) \\ \Psi_{\phi_{\mathrm{y}}}(\mathbf{z}_{\mathrm{d}}) \\ \Psi_{\dot{\gamma}}(\mathbf{z}_{\mathrm{d}}) \\ \Psi_{\ddot{\gamma}}(\mathbf{z}_{\mathrm{d}}) \\ \Psi_{\dot{x}_{\mathrm{tr}}}(\mathbf{z}_{\mathrm{d}}) \\ \Psi_{\ddot{x}_{\mathrm{tr}}}(\mathbf{z}_{\mathrm{d}}) \\ \Psi_{\dot{l}_{\mathrm{r}}}(\mathbf{z}_{\mathrm{d}}) \\ \Psi_{\ddot{l}_{\mathrm{r}}}(\mathbf{z}_{\mathrm{d}}) \end{bmatrix} \leq \begin{bmatrix} \phi_{\mathrm{x,max}} \\ \phi_{\mathrm{y,max}} \\ \dot{\gamma}_{\mathrm{max}} \\ \ddot{\gamma}_{\mathrm{max}} \\ \dot{x}_{\mathrm{tr,max}} \\ \ddot{x}_{\mathrm{tr,max}} \\ \dot{l}_{\mathrm{r,max}} \\ \ddot{l}_{\mathrm{r,max}} \end{bmatrix}. \tag{10e}$$

The optimization variables consist of a novel input $\mathbf{v} \in \mathbb{R}^3$ and the transition time t_{f}. The inputs $\mathbf{v}$

$$\mathbf{v} = \begin{bmatrix} X_{\mathrm{pl}}^{(5)} & Y_{\mathrm{pl}}^{(5)} & Z_{\mathrm{pl}}^{(5)} \end{bmatrix}^{\mathrm{T}} \tag{11}$$

represent the fifth derivative of each position coordinate. The cost (10a) consists of the free transition time t_{f}, the deviation of the payload position from the line containing $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$, and a regularization term with $\mathbf{R} = \mathbf{I}_3$. The scalar weight ω_{dist} serves as a tuning factor. The function

$$d = \frac{\left|\left(\mathbf{r}_{\mathrm{pl,d}}(t) - \mathbf{w}_i\right) \times \mathbf{e}_w\right|}{|\mathbf{e}_w|} \tag{12}$$

where $\mathbf{e}_w = \mathbf{w}_{i+1} - \mathbf{w}_i$, computes the distance between the position $\mathbf{r}_{\mathrm{pl,d}}(t)$ and the straight line containing the points $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$. The cost (10a) includes the quadratic distance d^2 to avoid the occurrence of square roots of decision variables, which otherwise leads to a significant increase of the computation time.

The corresponding discretized OCP is derived by introducing a fixed amount of computation intervals N, during which the inputs are assumed to be piecewise constant. The choice for N is crucial for the problem's scalability. In total, there are $1 + 3 \times N + 15 \times (N + 1)$ decision variables because we consider t_{f}, $\mathbf{v} \in \mathbb{R}^3$ at N points, and $\mathbf{z}_{\mathrm{d}} \in \mathbb{R}^{15}$ at $N + 1$ grid points.

The solution of the OCP is denoted by $\mathbf{v}^*(t)$, $t \in [t_0, t_{\mathrm{f}}^*]$, whereas the scalar optimal transition time is represented through t_{f}^*. The OCP computes a solution satisfying the first three requirements formulated in Section II-B. Because the corresponding payload trajectory $\mathbf{z}_{\mathrm{d}}^*(t)$, $t \in [0, t_{\mathrm{f}}^*]$ is obtained by integrating the discontinuous input $\mathbf{v}^*(t)$ with the integrator chain dynamics (10b), it holds that $\mathbf{r}_{\mathrm{pl,d}}(t) \in C^4$. Consequently, it follows that $\mathbf{r}_{\mathrm{pl,d}}^{(4)}(t) \in C^0$ and thus it yields $\mathbf{x}_{\mathrm{d}} \in C^1$ and $\mathbf{u}_{\mathrm{FF}} \in C^0$ according to (7) and (8). The OCP is solved for an arbitrary initial condition $\mathbf{z}_0$ denoted by (10c), whereas the terminal constraint (10d) is chosen such that the payload will be placed at waypoint $\mathbf{w}_{i+1}$. The velocity and acceleration constraints of the actuated degrees of freedom as well as sway angle constraints (2) are considered by reconstructing each variable with its parameterization in (10e). By tuning ω_{dist} properly, the geometrical distance of the resulting trajectory $\mathbf{z}_{\mathrm{d}}^*(t)$ to the desired path becomes sufficiently small.

Note that the parameterizations introduced in (10e) contain discontinuities at states, where $l_{\mathrm{r}} = 0$, $x_{\mathrm{tr}} = 0$, or $\phi_{\mathrm{x}}, \phi_{\mathrm{y}} \geq 90°$. The discontinuities can be excluded through adding additional constraints to (10) as it is done for the sway angles, which are limited by (3).

The notation

$$\left[t_{\mathrm{f}}^*, \mathbf{v}^*(t|t_0), \mathbf{z}_{\mathrm{d}}^*(t|t_0)\right] = \mathcal{O}(t_0, \mathbf{w}_i, \mathbf{w}_{i+1}, \mathbf{z}_0, \omega_{\mathrm{dist}}), \quad t \in \left[t_0, t_{\mathrm{f}}^*\right]$$

is introduced describing the solution variables and input parameters. Therein, it holds that $\mathbf{r}_{\mathrm{pl,d}}(t|t_0) = \mathbf{r}_{\mathrm{pl,d}}(t + t_0)$ emphasizing that the trajectory resulted from the OCP solved at the time step t_0. The input parameters required for solving the OCP (10) at a time step t_0 are the waypoints $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$ defining the straight line that the trajectory should approach, the initial condition $\mathbf{z}_0$, and the tuning parameter ω_{dist}.

IV. Trajectory Generation for Multiple Connection Lines

In order to satisfy (4) for the transitions between connection lines, the OCP (10) is employed multiple times in a novel trajectory generation algorithm. The basic procedure for an exemplary transition is explained in Section IV-A for the transition between three waypoints $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$. The overall algorithm and its properties is given in Section IV-B.

A. Transition Between Two Connection Lines

In a first step, the OCP

$$\left[{}^{\mathrm{cur}}t_{\mathrm{f}}^*, {}^{\mathrm{cur}}\mathbf{v}^*(\cdot|t_0), {}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^*(\cdot|t_0)\right] = \mathcal{O}_{\mathrm{cur}}(t_0, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}_0, \cdot)$$

is solved for the starting time step t_0 and the initial state $\mathbf{z}_0$, where $(\cdot)$ are placeholders for variables that are not relevant. The resulting trajectory describes the motion to the *currently* targeted waypoint, which is $\mathbf{w}_2$. The values ${}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^*(t_j|t_0)$ with $t_j \in [t_0, {}^{\mathrm{cur}}t_{\mathrm{f}}^*]$ on this trajectory can be utilized as initial condition for the OCP

$$\left[{}^{\mathrm{pre}}t_{\mathrm{f}}^*, {}^{\mathrm{pre}}\mathbf{v}^*(\cdot|t_0), {}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^*(\cdot|t_j)\right] = \mathcal{O}_{\mathrm{pre}}\left(t_j, \mathbf{w}_2, \mathbf{w}_3, {}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^*(t_j|t_0), \cdot\right).$$

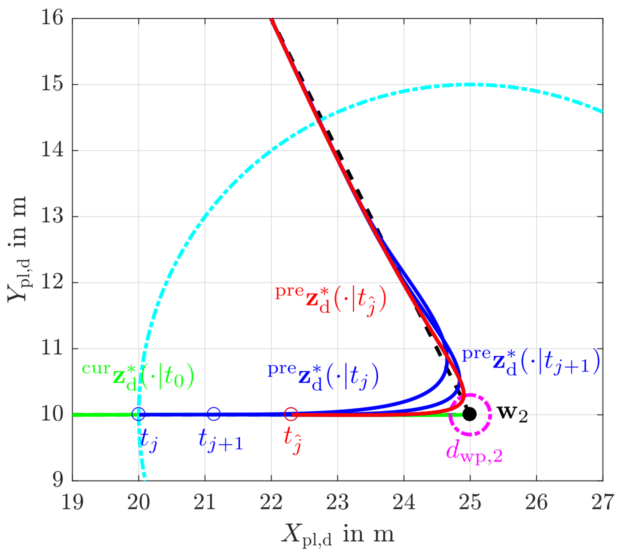

Fig. 3. Illustration of the introduced algorithm for generating a payload trajectory ${}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}$ between two connection lines of three waypoints.

Note that $\mathcal{O}_{\mathrm{pre}}$ computes the trajectory for the motion along the next connection line between $\mathbf{w}_2$ and $\mathbf{w}_3$ with an initial condition on the trajectory for $\mathbf{w}_1$ to $\mathbf{w}_2$. Thus, ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}$ implicitly contains the transition between the connection lines $\mathbf{w}_1$ to $\mathbf{w}_2$ and $\mathbf{w}_2$ to $\mathbf{w}_3$. The trajectory ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_j)$ is considered as *prediction*, because it is discarded for all t_j as long as

$$\exists t_{\hat{j}} \ : \ \exists \tau \ : \ \left| {}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\tau|t_{\hat{j}}) - [\mathbf{w}_2, 0, \ldots, 0]^{\mathrm{T}} \right| \leq d_{\mathrm{wp,2}} \tag{13}$$

is not satisfied.

Assuming that the predicted trajectory ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_{\hat{j}})$ satisfies (13) at the time step $t_{\hat{j}}$, the remainder of the currently planned trajectory ${}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(t|t_0)$, $t \in [t_{\hat{j}}, {}^{\mathrm{cur}}t_{\mathrm{f}}^{*}]$ is replaced with ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(t|t_{\hat{j}})$, $t \in [t_{\hat{j}}, {}^{\mathrm{pre}}t_{\mathrm{f}}^{*}]$. The currently planned trajectory is thus extended to

$${}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(t|t_0) := \begin{cases} {}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(t|t_0) & t \in [t_0, t_{\hat{j}}) \\ {}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(t|t_{\hat{j}}) & t \in \left[t_{\hat{j}}, {}^{\mathrm{pre}}t_{\mathrm{f}}^{*}\right]. \end{cases} \tag{14}$$

Similarly it is defined that ${}^{\mathrm{cur}}t_{\mathrm{f}}^{*} := {}^{\mathrm{pre}}t_{\mathrm{f}}^{*}$ and

$${}^{\mathrm{cur}}\mathbf{v}^{*}(t|t_0) := \begin{cases} {}^{\mathrm{cur}}\mathbf{v}^{*}(t|t_0) & t \in [t_0, t_{\hat{j}}) \\ {}^{\mathrm{pre}}\mathbf{v}^{*}(t|t_{\hat{j}}) & t \in \left[t_{\hat{j}}, {}^{\mathrm{pre}}t_{\mathrm{f}}^{*}\right]. \end{cases} \tag{15}$$

The procedure is illustrated in Fig. 3. The currently tracked trajectory ${}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_0)$, obtained through solving $\mathcal{O}_{\mathrm{cur}}$, is highlighted in green. The first predicted trajectory ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_j)$, resulting from $\mathcal{O}_{\mathrm{pre}}$, is computed during time step t_j. Because ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_j)$ neither intersects, or touches the purple circle with radius $d_{\mathrm{wp,2}}$, there exists no time step where the geometrical distance between the predicted trajectory and the second waypoint is smaller than the desired distance $d_{\mathrm{wp,2}}$. Hence, the predicted trajectory is discarded and the currently tracked trajectory is not changed. The same holds for the next time step t_{j+1} with the predicted trajectory ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_{j+1})$. During the time step $t_{\hat{j}}$, the predicted trajectory ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_{\hat{j}})$ intersects the purple circle and thus (13) is satisfied. The corresponding predicted trajectory is highlighted in red and replaces the remainder of the green trajectory.

Algorithm 1 Trajectory Gen. For Multiple Connection Lines

```
**Initial time step**
$[\cdot, {}^{\mathrm{cur}}\mathbf{v}^{*}(\cdot|t_0), {}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_0)] = \mathcal{O}_{\mathrm{cur}}(t_0, \mathbf{w}_1, \mathbf{w}_2, \mathbf{z}_0, \omega_{\mathrm{dist}})$
Set $i = 2$
**For every time step $t_j$ afterwards**
**if** $i \leq N_{\mathrm{wp}}$ **then**
    $[\cdot, \cdot, {}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_j)] = \mathcal{O}_{\mathrm{pre}}(t_j, \mathbf{w}_i, \mathbf{w}_{i+1}, {}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(t_j|t_0), \omega_{\mathrm{dist}})$
    **if** ${}^{\mathrm{pre}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_j)$ satisfies (13) **or** $i = N_{\mathrm{wp}}$ **then**
        Update ${}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_0)$, ${}^{\mathrm{cur}}\mathbf{v}^{*}(\cdot|t_0)$ as in (14) and (15)
        $i = i + 1$
    **end if**
**end if**
$j = j + 1$
```

B. Overall Algorithm

The whole process described in Section IV-A is repeated until $\mathcal{O}_{\mathrm{pre}}$ computes the trajectory ending at the last waypoint $\mathbf{w}_{N_{\mathrm{wp}}}$. Consequently, the current trajectory is extended one last time yielding the overall trajectory $\mathbf{z}_{\mathrm{d}}^{*}(t), \mathbf{v}^{*}(t), \ t \in [t_0, t_{\mathrm{f}}^{*}]$. The first three entries of $\mathbf{z}_{\mathrm{d}}^{*}(t)$ are denoted by $\mathbf{r}_{\mathrm{pl,d}}^{*}(t)$. The overall algorithm is the main result of this work and is summarized as Algorithm 1. Utilizing Algorithm 1 yields a trajectory ${}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(\cdot|t_0)$ in Cartesian coordinates. In order to obtain the feed-forward control input $\mathbf{u}_{\mathrm{FF}}$ and state reference $\mathbf{x}_{\mathrm{d}}$, the parameterizations (7) and (8) are utilized.

Note that if the OCP is feasible, it is guaranteed to find a solution satisfying (13) as long as $d_{\mathrm{wp,i+1}} > 0$ holds. Due to the terminal constraint of $\mathcal{O}_{\mathrm{cur}}$, it holds that

$$\exists \tau \ : \ \left| {}^{\mathrm{cur}}\mathbf{z}_{\mathrm{d}}^{*}(\tau|t_0) - [\mathbf{w}_{i+1}, 0 \ldots, 0]^{\mathrm{T}} \right| \leq d_{\mathrm{wp,i+1}}. \tag{16}$$

There are no issues with feasibility because the distance between connection lines and trajectory is only weighted in the cost. A disadvantage of only weighting this distance within the cost is that there is no fixed upper bound for the geometrical distance between trajectory and connection line except (13).

V. Simulation Results

This section discusses two exemplary payload trajectories resulting from the introduced OCP (10) and the method of Section IV utilizing two different values for ω_{dist}. The simulation results are obtained utilizing the *casadi*-plugin [24] in MATLAB 2019b. The optimization problem has been solved with *casadi.nlpsol* and employing the solver *ipopt*. Depending on the amount of time intervals N and the initial guess, the computation time for solving the OCP (10) in MATLAB varies roughly between 3 s ($N = 100$) and 82 s ($N = 5000$). The OCPs solved in this section are computed using $N = 1000$ equidistant time intervals, which vary in length because the transition time is a decision variable.

The two trajectories (colors yellow and blue) are computed for the three waypoints $\mathbf{w}_1 = [1.74, 11.72, -37]^{\mathrm{T}}$, $\mathbf{w}_2 = [15.08, -16.69, -36.46]^{\mathrm{T}}$, and $\mathbf{w}_3 = [8.8304, -20.02, -36.63]^{\mathrm{T}}$. The coordinates are not integer values, because this simulation setup is motivated by recorded measurement data. The initial state of the payload is chosen as $\mathbf{z}_0 = [\mathbf{w}_1, 0, \ldots, 0]^{\mathrm{T}}$. The desired minimum

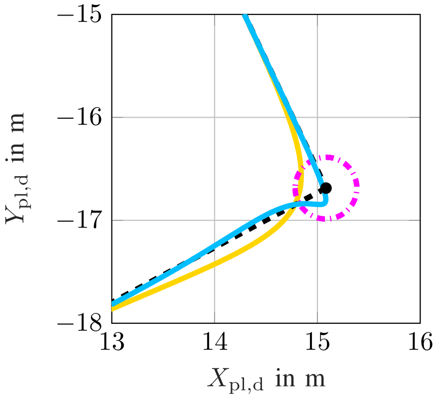

Fig. 4. Payload trajectory in the horizontal plane of a tower crane computed with Algorithm 1. When generating the blue trajectory, the path tracking error is weighted higher than when generating the yellow trajectory.

distance $d_{\mathrm{wp},2}$ is chosen to 0.3 m and indicated by a purple circle. Because it is unnecessary to predict trajectories right at the start, $\mathcal{O}_{\mathrm{pre}}$ is only evaluated for positions in a 3 m radius around $\mathbf{w}_2$. The solution of $\mathcal{O}_{\mathrm{cur}}$ is computed with $\omega_{\mathrm{dist}} = 5$ in both cases. However, the predictions are computed utilizing $\omega_{\mathrm{dist,yellow}} = 0.02$ and $\omega_{\mathrm{dist,blue}} = 0.1$. Thus, the squared geometrical distance d^2 is more expensive during the generation of the blue trajectory than during the generation of the yellow one.

The simulated geometrical paths of the trajectories during the transition between the connection lines are shown in Fig. 4. Because the squared distance d^2 is more weighted for the blue lines, the geometrical path of the blue trajectory directly approaches the connection line of $\mathbf{w}_2$ and $\mathbf{w}_3$ accepting an overshoot. On the other hand, the yellow trajectory performs a smooth transition in the shape of an arc, which is time efficient but yields larger squared distances d^2.

If ω_{dist} is chosen too large, the transition looks like the blue trajectory in Fig. 4 or may even contain loops. However, if ω_{dist} is chosen too small, a smooth and time-efficient transition is computed but the path error during the movement along the connection line is larger. The choice of ω_{dist} depends on the width of the transport corridors that are available at the construction site. It is important to choose ω_{dist} sufficiently high such that the payload trajectory remains in the corridors. If this is the case, the preferable choice is given by the lower bound of all weights that yield maximum path errors remaining in the width of the corridors. Then, the most efficient solution is obtained satisfying the geometrical constraints on the construction site while minimizing the overall transition time. The transition time t_{f}^* of the overall trajectory $\mathbf{r}^*_{\mathrm{pl,d}}(t)$ and maximum path error

$$e_{\max} = \max_{\tau\in[0,t_{\mathrm{f}}^*]}\left\{\min\left\{d\left(\mathbf{r}^*_{\mathrm{pl,d}}(\tau),\mathbf{w}_1,\mathbf{w}_2\right), d\left(\mathbf{r}^*_{\mathrm{pl,d}}(\tau),\mathbf{w}_2,\mathbf{w}_3\right)\right\}\right\} \tag{17}$$

have been computed for multiple values

$$100\,\omega_{\mathrm{dist}} \in \{2, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 10, 12.5, 15\}. \tag{18}$$

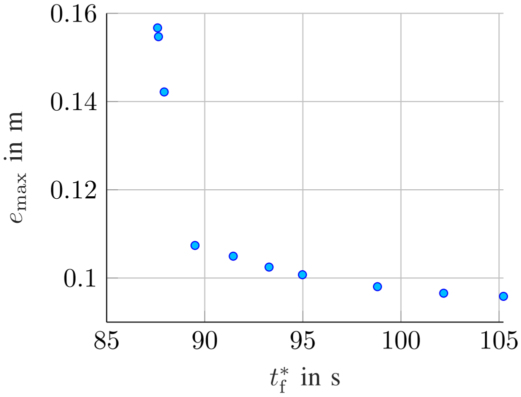

Fig. 5. Overall transition time t_{f}^* and maximum path error $e_{\max}$ obtained through solving the prediction steps of the simulation scenario with different values of the weight ω_{dist}.

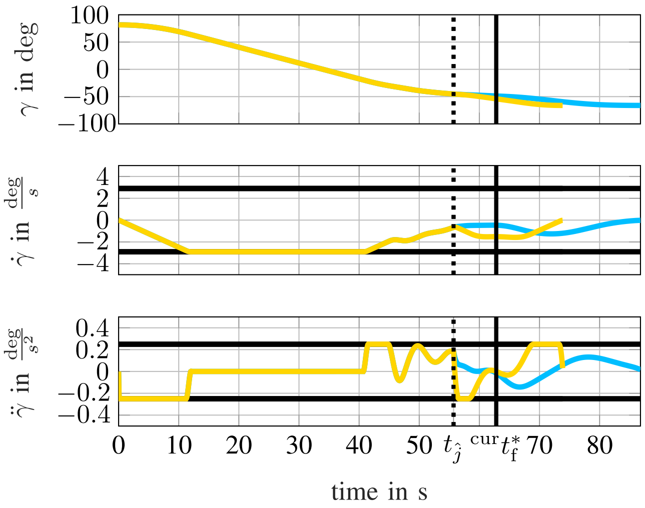

Fig. 6. Trajectory for the slewing angle γ_{d} of a tower crane computed with Algorithm 1. The path tracking error is weighted higher during the generation of the blue trajectory than during the generation of the yellow trajectory.

Fig. 5 shows, e.g., $\omega_{\mathrm{dist}} = 0.045$ is the most efficient solution if the maximum allowed path error is 12 cm.

The overall trajectories for the slewing angle and trolley position are shown in Figs. 6 and 7. The trajectories of both actuated degrees of freedom are continuous and satisfy the respective constraints during every time step. Due to the inefficient transition of the blue trajectory, the transition time $t^*_{\mathrm{f,blue}} \approx 86.8$ s is longer than $t^*_{\mathrm{f,yellow}} \approx 73.8$ s. In comparison, the transition time obtained for stopping at waypoint $\mathbf{w}_2$ and moving the payload to $\mathbf{w}_3$ from steady state is $t_{\mathrm{f}}^* \approx 81.1$ s. The gain of the efficient transition in this example is thus a time save of 7.3 s, which is even larger for larger distances $d_{\mathrm{wp},2}$ and adds up for multiple transitions.

VI. Application to an Experimental Tower Crane

This section discusses the integration of the trajetory resulting from Algorithm 1 for a set of waypoints $\mathcal{W}$ to a large-scale tower crane in Section VI-A as well as the corresponding measurement results in Section VI-B. The waypoints do not consider the elasticity of the crane.

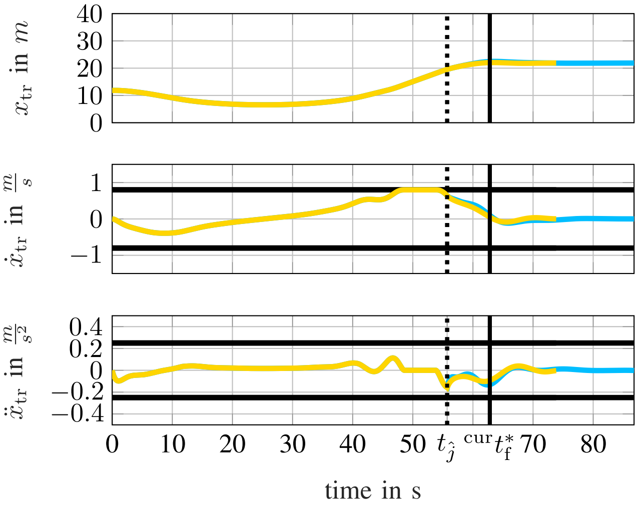

Fig. 7. Trajectory for the trolley position $x_{\mathrm{tr,d}}$ of a tower crane computed with the Algorithm 1. The path tracking error is weighted higher during the generation of the blue trajectory than during the generation of the yellow trajectory.

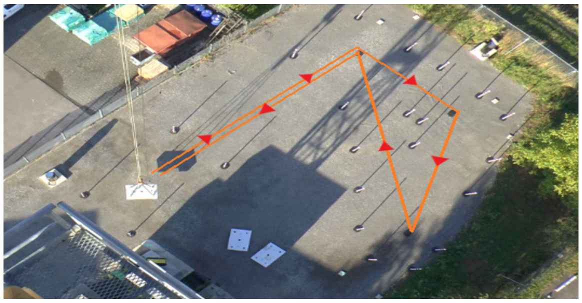

Fig. 8. Experimental obstacle course for the tower crane.

A. Experimental Setup

The considered experimental tower crane is a Liebherr 154 EC-H Litronic 6 tower crane shown in Fig. 1, for which a trajectory tracking controller was developed in [23]. The experimental tower crane has a hook height of 41 m and a jib length of 60 m. The attached payload has a weight of $m_{\mathrm{pl}} = 1$ t. The connection lines are marked through pylons that are deployed next to the connection lines as shown in Fig. 8. The waypoints $\mathcal{W} = \{A, B, C, D, B, A\}$ are shown in Fig. 9. The offline computed trajectory and tracking controller are implemented on a dSpace MicroAutoBox (MAB) II prototyping system. The amount of time intervals is chosen to $N = 1000$. The required computational time of the trajectory is 55.85 s. The overall trajectory $\mathbf{z}_{\mathrm{d}}^{*}$ includes nonequidistant time steps. Thus, the trajectory is interpolated accordingly before it is saved.

B. Measurement Results

The measurement results are obtained by moving the payload to the waypoint A and playing the offline computed trajectory. Fig. 9 shows the horizontal payload position of the trajectory and the payload position reconstructed with the estimated sway angles. During the transition from waypoint A to D, the path error remains smaller than 40 cm until the waypoint D is reached and the payload transitions smoothly between the connection lines. However, the tracking controller causes a loop during the transition at waypoint D and on the way back at waypoint B. The overshoot caused at waypoint D results from the tracking error of ϕ_{x} at $t = 60$ s and the tracking error of ϕ_{y} at $t = 80$ s, whereas the overshoot at waypoint B is caused by the error of ϕ_{x} at $t = 110$ s yielding a global maximum for the payload position error of 0.75 m. There are two major issues that cause these trajectory tracking errors. First, the dynamics considered in the OCP (10) model a rigid tower crane with a single pendulum. The experimental tower crane is characterized by a double pendulum and the elasticity of the tower crane's steel structure has an impact on the sway dynamics. Smaller tower cranes have less elasticity and thus the corresponding tracking performance is better. Second, the tracking controller on its own shows small path errors as mentioned in [23]. Further experiments showed a payload positioning accuracy of 0.5 m. The related scattering

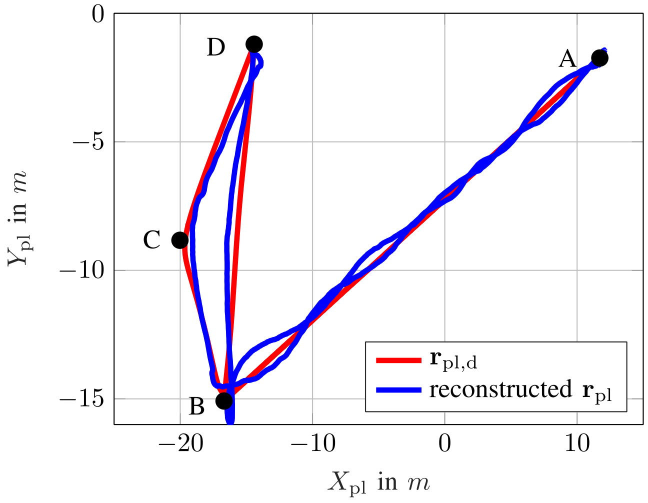

Fig. 9. Horizontal payload position trajectory (red) and real payload position reconstructed with the sway angles of an observer (blue) during the trajectory tracking with the experimental tower crane.

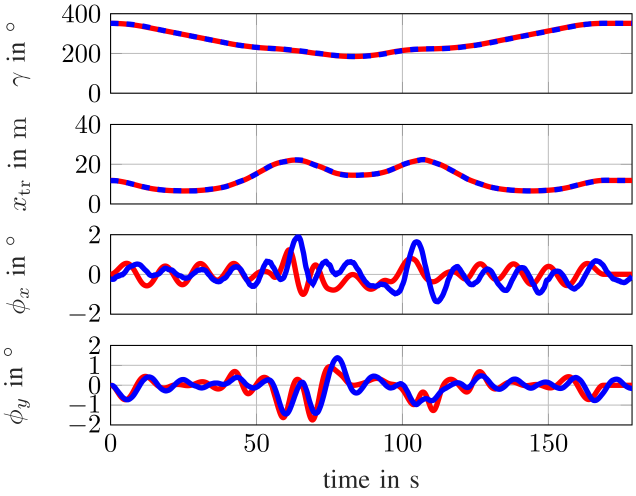

Fig. 10. State reference (red) and recorded states (blue) of the experiment. The slewing angle and trolley position are measured by sensors, whereas the sway angles ϕ_{x} and ϕ_{y} are estimated by an observer.

is also supported by disturbances such as wind acting on the jib. Both reasons lead to a tracking error of the sway angles, whereas the positions of the actuated degrees of freedom show negligible tracking errors, which is shown in Fig. 10.

A video comparing the trajectory obtained by concatenating the solution of the OCP (10) for each straight line and the trajectory resulting from Algorithm 1 is available at https://www.youtube.com/watch?v=VHkrDUoDKQU.

VII. Conclusion

This brief introduces an algorithm for generating a payload position trajectory such that the payload follows straight connection lines of a given set of waypoints and for computing smooth transitions at the intersection of the connection lines. The requirements on the payload position trajectory are that it needs to be four times continuously differentiable, the velocity and acceleration constraints on the tower crane's actuated degrees of freedom and sway angles need to be satisfied, the geometrical path needs to be sufficiently close to the straight connection lines between the waypoints, and the transitions between the connection lines are smoothed such that the distance between the payload position and a waypoint falls below a user-defined minimum distance. The smoothed transitions yield time efficiency because the payload is not stopped and re-accelerated at each waypoint. The trajectory between two waypoints is computed by solving an OCP for the payload position in Cartesian coordinates. The novel algorithm then utilizes the OCP for predicting trajectories to the next waypoint while taking the values of the trajectory to the currently targeted waypoint as initial conditions. Once the predicted trajectory satisfies all requirements at a time step, the predicted trajectory then becomes the currently planned trajectory. The application to a real tower crane showed that an obstacle course consisting of line segments is passed.

Room for improvement of the algorithm is given by changing or adding constraints to the OCP that bound the path error between the payload position trajectory and the connection line, which is not explicitly bounded yet. The performance of large-scale tower cranes with elastic structure can be further improved by considering elastic degrees of freedom in the underlying dynamics. This yields a different flat output and thus the state and input parameterizations deduced in the preliminaries of this work do not hold anymore.

References

[1] N. Sun, Y. Fang, Y. Zhang, and B. Ma, "A novel kinematic coupling-based trajectory planning method for overhead cranes," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 166–173, Feb. 2012.

[2] H. Chen, Y. Fang, and N. Sun, "Optimal trajectory planning and tracking control method for overhead cranes," *IET Control Theory Appl.*, vol. 10, no. 6, pp. 692–699, Apr. 2016.

[3] Z. Wu and X. Xia, "Optimal motion planning for overhead cranes," *IET Control Theory Appl.*, vol. 8, no. 17, pp. 1833–1842, Nov. 2014.

[4] H. Chen, Y. Fang, and N. Sun, "A swing constrained time-optimal trajectory planning strategy for double pendulum crane systems," *Nonlinear Dyn.*, vol. 89, no. 2, pp. 1513–1524, Jul. 2017.

[5] Z. Tian, L. Yu, H. Ouyang, and G. Zhang, "Swing suppression control in tower cranes with time-varying rope length using real-time modified trajectory planning," *Autom. Construct.*, vol. 132, Dec. 2021, Art. no. 103954.

[6] H. Ouyang, Z. Tian, L. Yu, and G. Zhang, "Motion planning approach for payload swing reduction in tower cranes with double-pendulum effect," *J. Franklin Inst.*, vol. 357, no. 13, pp. 8299–8320, Sep. 2020.

[7] G. Li, X. Ma, Z. Li, and Y. Li, "Time-polynomial-based optimal trajectory planning for double-pendulum tower crane with full-state constraints and obstacle avoidance," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 2, pp. 919–932, Apr. 2023.

[8] Z. Liu, N. Sun, Y. Wu, H. Chen, X. Liang, and Y. Fang, "Multi-objective trajectory planning with state constraints for 5-DOF underactuated tower crane systems," in *Advances in Applied Nonlinear Dynamics, Vibration and Control*, X. Jing, H. Ding, and J. Wang, Eds. Singapore: Springer, 2022, pp. 710–728.

[9] M. Thomas, J. Qiu, and O. Sawodny, "Trajectory sequence generation and static obstacle avoidance for automatic positioning tasks with a tower crane," in *Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2021, pp. 1–6.

[10] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, *Path Planning and Trajectory Planning Algorithms: A General Overview*. Cham, Switzerland: Springer, 2015, pp. 3–27.

[11] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Cham, Switzerland: Springer, 2008.

[12] M. Egerstedt and C. F. Martin, "Optimal trajectory planning and smoothing splines," *Automatica*, vol. 37, no. 7, pp. 1057–1064, Jul. 2001.

[13] W. Van Loock, G. Pipeleers, and J. Swevers, "B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction," *Mech. Sci.*, vol. 6, no. 2, pp. 163–171, Sep. 2015.

[14] J. A. De Doná, F. Suryawan, M. M. Seron, and J. Lévine, *A Flatness-Based Iterative Method for Reference Trajectory Generation in Constrained (NMPC)*. Berlin, Germany: Springer, 2009, pp. 325–333.

[15] F. Stoican, I. Prodan, and D. Popescu, "Flat trajectory generation for way-points relaxations and obstacle avoidance," in *Proc. 23rd Medit. Conf. Control Autom. (MED)*, Jun. 2015, pp. 695–700.

[16] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "On differentially flat nonlinear systems," in *Nonlinear Control Systems Design* (IFAC Symposia Series), M. Fliess, Ed. New York, NY, USA: Pergamon, 1993, pp. 159–163.

[17] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, Jun. 1995.

[18] H. Sira-Ramirez and S. K. Agrawal, *Differentially Flat Systems*. Boca Raton, FL, USA: CRC Press, 2004.

[19] J. Oldenburg and W. Marquardt, "Flatness and higher order differential model representations in dynamic optimization," *Comput. Chem. Eng.*, vol. 26, no. 3, pp. 385–400, Mar. 2002.

[20] M. Böck and A. Kugi, "Real-time nonlinear model predictive path-following control of a laboratory tower crane," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1461–1473, Jul. 2014.

[21] G. G. Rigatos, *Nonlinear Control and Filtering Using Differential Flatness Approaches: Applications to Electromechanical Systems* (Studies in Systems, Decision and Control). Springer, 2015. [Online]. Available: https://books.google.de/books?id=J3zMCQAAQBAJ

[22] K. L. Knierim, K. Krieger, and O. Sawodny, "Flatness based control of a 3-DOF overhead crane with velocity controlled drives," *IFAC Proc. Volumes*, vol. 43, no. 18, pp. 363–368, 2010.

[23] F. Rauscher and O. Sawodny, "Modeling and control of tower cranes with elastic structure," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 64–79, Jan. 2021.

[24] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.