Flatness-based Feedforward Control of a Stacker Crane with Online Trajectory Generation

Charlotte Tkany, Martin Grotjahn
Institute of Engineering Design, Mechatronics and Electromobility
Hochschule Hannover
Hannover, Germany
charlotte.tkany@hs-hannover.de,
martin.grotjahn@hs-hannover.de

Johannes Kühn
RDS Motion Control
Lenze Automation GmbH
Braunschweig, Germany
johannes.kuehn@lenze.com

***Abstract*—Flatness-based feedforward control is an approach for combining fast motion with low oscillations for nonlinear or flexible drive systems. Its desired trajectories must be continuously differentiable to the degree of the system order. Designing such trajectories, that also reach the dynamic system limits, poses a challenge. Common solutions, like Gevrey functions, usually require lengthy offline calculations. To achieve a quicker and simpler industrial-suited solution, this paper presents a new online trajectory generation scheme. The algorithm utilizes higher order s-curve trajectories created by a cyclic filtering process using moving average filters. An experimental validation proves the capability as well as industrial applicability of the presented approach for flexible structures like stacker cranes.**

Keywords: flatness-based control; flexible structure; online trajectory generation; moving average filter; increasing continuous differentiability; dynamic trajectories

I. INTRODUCTION

In the last decades the intralogistics sector has grown significantly due to the e-commerce boom. In 2017 the top 20 world-wide suppliers of material handling systems (e.g. stacker cranes) exhibited a universal revenue growth for the first time in over two decades [1].

Stacker cranes (see Fig. 1) are characterised by a tall mast and a load handling device, denoted LHD. The LHD can be moved vertically along the mast and extended outwards to deposit or remove elements from a rack. In this paper, belt driven stacker cranes are examined. Such stacker cranes consist of a drive motor connected to a belt via an Omega drive. The motor is fixed to the moving platform of the crane.

The mast of a stacker crane is liable to oscillations whenever it moves, which puts a mechanical strain on the construction. Some form of oscillation reduction needs to be applied to guarantee safety during dynamic motion. Additionally, the LHD can only be safely extended after oscillations subside. Therefore, a significant reduction of mast oscillations is critical in order to achieve a time efficient operation of the stacker crane. The maximum improvement can be achieved, if the oscillation reduction scheme is combined with dynamic positioning of the stacker crane. One main difficulty in reducing mast oscillations is the dependency of mast eigenfrequency on LHD height. Thus, a time-variant oscillation control problem results. If a limitation of mast oscillations can be guaranteed, construction of lighter, less stiff stacker cranes becomes possible.

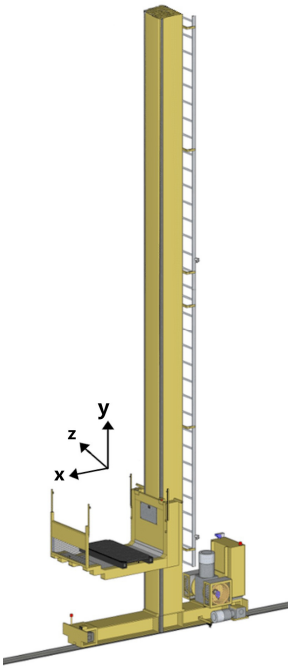

Figure 1: Single mast stacker crane [17]

One solution is the reduction of mast oscillations through advanced trajectory planning and control approaches. Many approaches have been examined, like state space control [2], passivity based control [3], time-optimal trajectory planning [3] or input shaping [4]. One increasingly popular option is the development of a flatness-based open-loop control as employed in [2], [3], [5] and [6]. However, only [3] has validated its approach for a simultaneous motion of platform and LHD, thus addressing the issue of the time-variant mast eigenfrequency. Other papers (e.g. [2], [5]) have included LHD motion in modelling, but not in the final flatness-based approach. In [2], the LHD is assumed to be at final height for the entire motion. Since the corresponding control validation does not include simultaneous motion of

platform and LHD, the validity of this simplification cannot be judged.

The flatness-based approach has also proven effective in reducing load side vibrations of any number of different flexible motion systems (see e.g. [4], [7], [8]). But of these only [4] includes simultaneous motion of two axis similar to the motion of platform and LHD.

Contrarily to the cited publications, the time-variant mast eigenfrequency will be considered during development and validation of the flatness-based feedforward control scheme presented in this paper.

Another important aspect of flatness-based control is the design of appropriate trajectories. In order to receive a continuous output, the desired trajectory has to have a number of smooth derivatives corresponding to system order n, making it a function of (differentiability) class C^n. For systems with an order higher than three, the design of such a trajectory is non-trivial.

Many different types of trajectories have been explored in the literature, such as polynomial functions ([7]) and Gevrey functions ([2], [5]). Newer approaches include the use of classical s-curve trajectories, whose differentiability is increased through various means ([4], [6], [8]). This will be expanded upon in section III.

This paper focuses on an additional requirement, which is the implementation on a real time system. This can be achieved through either online trajectory generation or reliance on offline calculations that can be accessed online through a look-up table or similar means.

Online trajectory generation stipulates a generation that is self-contained within every control cycle, generally every millisecond. Also, a seamless transition from one trajectory to another has to be possible, meaning that no significant delay occurs before the start of a new motion.

This generally requires a low computational complexity, which many approaches cannot achieve. In that case, additional offline calculations become necessary. This requires trajectories for every single possible motion, with every possible set of boundary conditions to be calculated offline before the control can be employed. The number of possible motions often consists of several hundred thousand or even over a million possible combinations [9]. This requires a deep analysis of the possible demands and time-consuming calculations before the control can be employed.

This aspect is rarely explored explicitly in literature, other than for optimization-based approaches, whose computational complexity always requires offline calculations.

In [2] it is mentioned that online implementation should theoretically be possible for the presented Gevrey function-based approach. They recommend exploration in further research. Yet to the authors knowledge no such follow up exist, leading to the conclusion that every beforementioned trajectory planning approach requires some form of offline calculation.

This is relevant since online trajectory generation plays into the already existing advantage of a feedforward-based oscillation reduction approach. Since such an approach can be combined with any number of pre-existing control schemes, it can be integrated retroactively into many systems. This makes it an especially simple and industrial-suited approach. A requirement of lengthy offline calculations and analysis would negate this advantage.

Therefore, the aim of this paper is the development of a flatness-based oscillation reduction scheme with online trajectory generation.

The online implementation puts a constraint on the computational complexity, which is increased through the inclusion of a time-variant eigenfrequency. Therefore, a relatively simple multi-body modelling approach is used, in contrast to a continuous modelling approach based on the Euler-Bernoulli beam theory (see [2], [3], [5], [10]). This modelling is presented in section II in conjunction with the introduction of the test bed, which serves as an example of a miniature stacker crane.

With regards to the trajectory generation, the low computational cost is achieved through a cyclic calculation approach, that distributes the cost across the entire motion, as presented in section III. It also makes use of dynamic s-curve trajectories, whose differentiability is increased through the use of moving average filters.

Finally, the derived algorithm is experimentally validated in section IV.

II. Modelling of the Test Bed

The test bed consists of a platform with an approximately 1.2 m long mast running along a belt on a 3 m rail (see Fig. 2). A LHD can be moved up and down the mast via a belt and its motor, positioned on the platform. The platform is driven by an Omega drive, visible in Fig. 2. The test bed was designed to have a realistic oscillation behaviour, meaning that the eigenfrequencies of mast and belt closely resemble eigenfrequencies of real stacker cranes. For the belt, this means that the stiffness of a much longer belt has to be approximated. Therefore, flat springs are connected to the belt in series on either end of the rail, as seen in Fig. 2.

The test bed is modelled by three masses, i.e. drive motor,

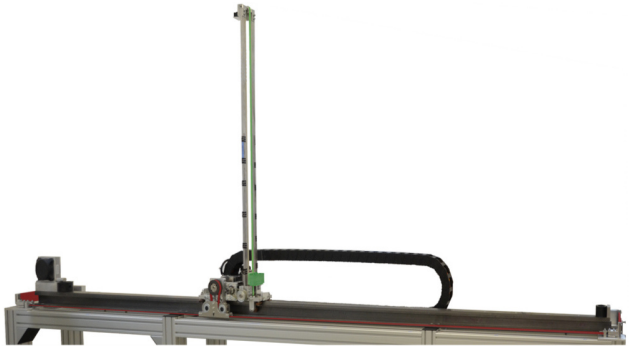

Figure 2: Test Bed, with the belt and flat springs highlighted in red and the LHD and its belt highlighted in green.

platform and mast, which are coupled by two springs, belt and mast, leading to a system with order $n = 6$. One can define a theoretical motor position x_1 in addition to platform position x_2, even though the motor is located on the platform. The motor position is calculated only from actual motor rotation. The platform position additionally includes the influence the belt elasticity has on the position (see Fig. 3). The third position of interest is load position x_3, which describes the horizontal position of the mast head. Following these definitions, belt and mast oscillations can be calculated as the difference between x_1, x_2 and x_2, x_3 respectively.

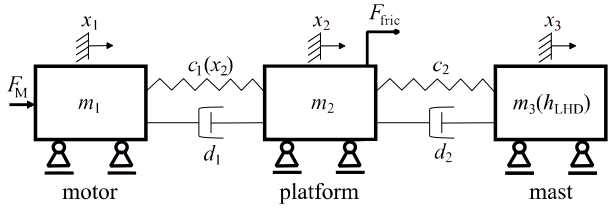

Figure 3: Mass-spring-damper model of the test bed

Following this modelling approach the mast is approximated as a massless beam with a point mass on top (see Fig. 4, right side), which is connected to a spring with stiffness c_2. This simple approximation is in contrast to often developed continuous models based on Euler-Bernoulli beam theory, as applied in [2], [3], [5] and [10].

The chosen simplification means, that only the first eigenfrequency is modelled. The consequences of this simplification will be examined during experimental validation in section IV.

Following this model, stiffness c_2 can be calculated by viewing the mast as a beam with length h which is clamped on one end (see Fig. 4, left side). When a force F is applied to the mast head, its resulting deflection can be calculated via [11]:

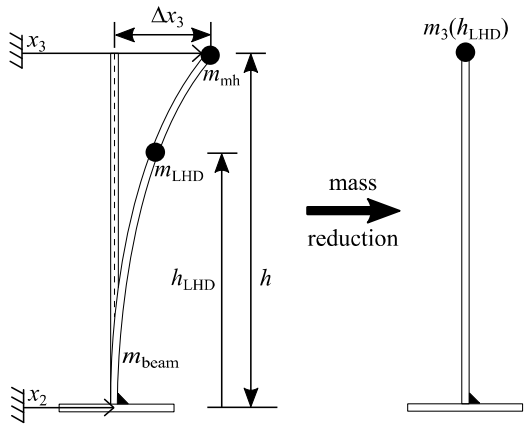

Figure 4: Mast model, with a detailed overview of all components on the left and the simplified model with one concentrated point mass on the right

$$\Delta x_3 = \frac{Fh^3}{3EI}, \tag{1}$$

with the mast's Young modulus E and its second moment of area I.

Applying Hooke's law, leads to the mast stiffness c_2 [11]:

$$c_2 = \frac{3EI}{h^3}. \tag{2}$$

Following the chosen model, the different mast masses have to be reduced into a single point mass at the mast head, as depicted in Fig. 4. This point mass must include the masses of the beam m_{beam}, the LHD m_{LHD} and the mast head m_{mh}. Only the latter enters into the resulting mass m_3 unchanged. For m_{beam} and m_{LHD} equivalent reduced point masses at the top of the mast (height h) have to be calculated:

$$m_3(h_{LHD}) = m_{beam,red} + m_{LHD,red}(h_{LHD}) + m_{mh}, \text{ with} \tag{3}$$

$$m_{beam,red} = \frac{33}{140} m_{beam} \tag{4}$$

$$m_{LHD,red}(h_{LHD}) = \frac{m_{LHD}}{4}\left(9\left(\frac{h_{LHD}}{h}\right)^4 - 6\left(\frac{h_{LHD}}{h}\right)^5 + \left(\frac{h_{LHD}}{h}\right)^6\right) \tag{5}$$

In (4) the mass of the continuous beam is concentrated into a point mass at the mast head. The factor 33/140 serves to keep the eigenfrequency constant through the transformation into an equivalent point mass system [12].

Equation (5) [13] reduces the mass of the LHD at current height h_{LHD} into an equivalent mass at height h.

For the resulting mass-spring system, the eigenfrequency can be calculated via:

$$f_{mast}(h_{LHD}) = \frac{1}{2\pi}\sqrt{\frac{c_2}{m_3(h_{LHD})}} \tag{6}$$

Since the reduced mass changes with LHD height, so does the frequency, leading to the earlier stated time-variant eigenfrequency. For this test bed the effective mass doubles, when the LHD is moved from the bottom to the top of the mast. For these LHD positions, the eigenfrequency changes inversely as depicted in Fig. 5.

Similarly, the belt eigenfrequency is also time-variant, since belt stiffness c_1 depends on platform position x_2. Effectively, the platform splits the belt into two parts, which can be approximated by parallel springs, leading to an increased stiffness towards the ends of the belt. Each parallel

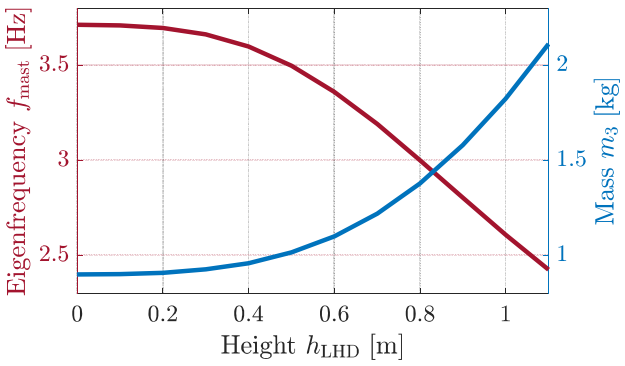

Figure 5: Mast eigenfrequency and mass dependent on LHD height

spring consists of a segment of the belt and its flat spring connected in series.

Friction force F_{fric} acting on the platform (see Fig. 3) is modelled using a non-linear Coulomb and viscous friction model. The system input is the motor force F_{M} and its state space vector is

$$\boldsymbol{x} = [x_1, \dot{x}_1, x_2, \dot{x}_2, x_3, \dot{x}_3]^{\mathrm{T}} . \tag{7}$$

This system can be expressed mathematically through the following three differential equations. The time dependency of the system states, i.e. $x_1(t)$, is not explicitly written to increase readability:

$$m_1\ddot{x}_1 = c_1(x_2)(x_2 - x_1) + d_1(\dot{x}_2 - \dot{x}_1) + F_{\mathrm{M}} \tag{8}$$

$$\begin{aligned} m_2\ddot{x}_2 = &-c_1(x_2)(x_2 - x_1) - d_1(\dot{x}_2 - \dot{x}_1) \\ &+ c_2(x_3 - x_2) + d_2(\dot{x}_3 - \dot{x}_2) - F_{\mathrm{fric}}(\dot{x}_2) \end{aligned} \tag{9}$$

$$m_3(h_{\mathrm{LHD}})\ddot{x}_3 = -c_2(x_3 - x_2) - d_2(\dot{x}_3 - \dot{x}_2) \tag{10}$$

This model is the basis for the flatness-based control approach presented in the following section.

III. Flatness-Based Control Approach

Differential flatness is a concept first introduced by Fliess et al. in 1992 [14]. It represents a new modelling approach for non-linear systems. For linear systems, it equals the property of controllability.

A system is flat, if a flat output y_{f} exists. This means that every state and every control variable can be expressed just via the flat output y_{f} and a finite number of its derivatives. That leads to the following relationships for a SISO system

$$\boldsymbol{x} = \boldsymbol{f}\left(y_{\mathrm{f}}, \dot{y}_{\mathrm{f}}, \ddot{y}_{\mathrm{f}}, \ldots, y_{\mathrm{f}}^{(n-1)}\right), \tag{11}$$

$$u = g\left(y_{\mathrm{f}}, \dot{y}_{\mathrm{f}}, \ddot{y}_{\mathrm{f}}, \ldots, y_{\mathrm{f}}^{(n)}\right), \tag{12}$$

where n denotes the system order. Thus, for a flat system the existence of a mapping between input and flat output is guaranteed. This transformation offers a way of inverting the system, without having to solve differential equations. That is why flatness-based approaches are often used in open-loop control. In theory, an oscillation free load side positioning can be achieved, if the load position is a flat output. An oscillation free desired trajectory has to be defined for the flat output and transformed into the corresponding reference trajectory using the flatness-based inverse system model. However, this assumes that the system model is 100 % accurate, which can never be achieved in practice.

For non-linear systems, there is no analytical method that discerns flat outputs. A suspected flat output can be verified through repeated differentiation and subsequent transformation of the results into the form of (11) and (12). For each flat system an infinite number of flat outputs exists. Therefore, it is desirable to choose a flat output which has a physical meaning. The design of appropriate trajectories is simplified, if the flat output is also the system output. Otherwise the internal dynamics of these two outputs have to be modelled and considered.

A. Flatness-based modelling of the test bed

With regard to the stacker crane, mast position x_3 is desired to be the flat output y_{f}. In order for x_3 to be a flat output of the system the damping has to be neglected from (8), (9) and (10). Since both mast and belt are only weakly damped, this is an acceptable simplification.

From the resulting system the reference trajectories x_1 and $\dot{x}_1$ have to be calculated as well as the motor force F_{M}, which is applied as the feedforward term. This can be achieved by repeated differentiation of the flat output and transformation of the resulting equations. Due to their time-variant nature, derivatives of both c_1 and m_3 have to be calculated as well. This is neglected in the following simplified version, which treats both parameters as time-invariant:

$$\begin{aligned} F_{\mathrm{M}} = &\frac{m_1 m_2 m_3}{c_1 c_2} y_{\mathrm{f}}^{(6)} \\ &+ \left(\frac{m_1(m_2 + m_3)}{c_1} + \frac{m_3(m_1 + m_2)}{c_2}\right) y_{\mathrm{f}}^{(4)} \\ &+ (m_1 + m_2 + m_3) y_{\mathrm{f}}^{(2)} + F_{\mathrm{fric}}(t) + \frac{m_1}{c_1}\dot{F}_{\mathrm{fric}}(t) \end{aligned} \tag{13}$$

$$x_1 = y_{\mathrm{f}} + \left(\frac{m_2 + m_3}{c_1} + \frac{m_3}{c_2}\right) y_{\mathrm{f}}^{(2)} + \frac{m_2 m_3}{c_1 c_2} y_{\mathrm{f}}^{(4)} + \frac{F_{\mathrm{fric}}(t)}{c_1} \tag{14}$$

$$\dot{x}_1 = y_{\mathrm{f}}^{(1)} + \left(\frac{m_2 + m_3}{c_1} + \frac{m_3}{c_2}\right) y_{\mathrm{f}}^{(3)} + \frac{m_2 m_3}{c_1 c_2} y_{\mathrm{f}}^{(5)} + \frac{\dot{F}_{\mathrm{fric}}(t)}{c_1} \tag{15}$$

If m_3 is modelled as time-variant, the calculation of F_{M} changes to:

$$
\begin{aligned}
F_{\mathrm{M}} = & \frac{m_1 m_2 m_3}{c_1 c_2} y_{\mathrm{f}}^{(6)} + \frac{m_1 m_2 \dot{m}_3}{c_1 c_2} y_{\mathrm{f}}^{(5)} + \left(\frac{m_1 (m_2 + m_3)}{c_1} \right. \\
& \left. + \frac{m_3 (m_1 + m_2)}{c_2} + \frac{2 m_1 m_2}{c_1 c_2} \left(4 \frac{\dot{m}_3^2}{m_3} + 3 \ddot{m}_3 \right) \right) y_{\mathrm{f}}^{(4)} \\
& + \left(m_3 \left(\frac{m_1 + m_2}{c_2} + \frac{m_2}{c_1} \right) \right. \\
& \left. + \frac{3 m_1 m_2}{c_1 c_2} \left(\frac{\dot{m}_3^3}{m_3^2} - \frac{2 \dot{m}_3 \ddot{m}_3}{m_3} + m_3^{(3)} \right) \right) y_{\mathrm{f}}^{(3)} \\
& + \left(m_1 + m_2 + m_3 + \ddot{m}_3 \left(\frac{m_1 + m_2}{c_2} + \frac{m_2}{c_1} \right) \right. \\
& \left. + \frac{m_1 m_2}{c_1 c_2} m_3^{(4)} \right) y_{\mathrm{f}}^{(2)} + F_{\mathrm{fric}} + \frac{m_1}{c_1} \ddot{F}_{\mathrm{fric}}
\end{aligned} \tag{16}
$$

It should be noted that in addition to the system states both m_1 and F_{fric} are now dependent on the time t, which is once again not explicitly written to increase readability.

It becomes clear, that incorporating derivatives of even a single time-variant parameter exponentially increases complexity. Since high computational complexity is disadvantageous for the desired application, the impact of possible simplifications is tested through simulation. As one option all derivatives of m_3 are disregarded (as in (13)), while still treating the parameter m_3 itself as time-variant. A comparison of this option with (16) shows that the simplification only has a minimal impact on the resulting trajectories.

The impact can be evaluated by the maximum deviation that results for the force and reference trajectory as a consequence of the simplification. The simulation using (16) serves as reference for the calculated deviation. In Tab. I the maximum relative error, meaning maximum deviation divided by trajectory maxima, is shown for both the force and reference trajectory. The error for the reference trajectory is clearly negligible, while for the force a noticeable deviation occurs. Since the motor force merely acts as a feedforward term, any error of this size can easily be corrected by the applied follow-up control. The process is repeated for c_1, were the relative error is even smaller.

TABLE I: MAXIMUM RELATIVE ERROR CAUSED BY THE NEGLECTION OF THE DERIVATIVES OF m_3

	Maximum relative error [%]
$\boldsymbol{F_{\mathrm{M}}}$	1.5
$\boldsymbol{x_1}$	0.00125

Based on these results, for the final feedforward control, the derivatives of all time-variant parameters are neglected, while still treating the parameters themselves as time-variant. Meaning that (13), (14) and (15) are used, with c_1 and m_3 updated in every control cycle.

B. Flatness-based trajectory planning

In accordance with (12) the desired trajectory for a flat system has to be at least n-times continuously differentiable to result in a continuous system input. Flatness-based trajectory generation has to create a trajectory meeting this condition. Additionally, the trajectory generation has to produce all n derivatives of the desired trajectory, since they are required in (13), (14) and (15).

The most commonly used function types to fulfil the requirement of differentiability are polynomial and Gevrey functions. The latter is common because it a C^∞-function. Therefore, it is used in conjunction with continuous modelling approaches. However, it requires a non-trivial integral to be solved for every new trajectory, which entails a high computational cost. The authors are not aware of a successful online implementation of trajectory generation using a Gevrey function for flatness-based control.

With regards to time efficient positioning, s-curve trajectories with trapezoid acceleration profiles are very desirable. In contrast to polynomial functions, such trajectories can maintain a constant velocity or acceleration for a prolonged period of time. In classic trajectory generation such 3rd order s-curve profiles are very common. Unfortunately, an analytic calculation is only possible up to C^3. After that the complexity of the calculation becomes impractical ([15]), which is a hindrance for the current application.

In [4] cubic splines where fitted to a trapezoidal acceleration profile and twice integrated in order to retain the time efficient positioning while also increasing differentiability. This has the downside of requiring a cumbersome iterative trajectory generation process to match boundary conditions.

In [8] convolution is applied repeatedly to increase the differentiability of a trajectory, starting with a rectangular velocity profile. Convolution requires knowledge of the entire trajectory and repeated use entails a significant computational cost, which hinders once again an online implementation. Its superiority to polynomial functions with regards to positioning times was proven in [8] for C^4-trajectories.

This paper improves upon this approach by repeatedly applying a moving average filter to a s-curve trajectory in order to increase its differentiability until a C^6-trajectory results. Since it is a causal filter, the moving average does not require future knowledge of the trajectory. This enables the use of an algorithm that is repeated in each control cycle and only calculates the next values of the trajectory and its derivatives. That means, that the computational cost of trajectory generation is spread out over the duration of the motion. This drastically reduces the calculations required each cycle, allowing for online execution.

A moving average is calculated for a discrete time series via

$$M^{(m)}(t) = \frac{1}{m}\sum_{i=0}^{m-1} x(t-i)\,, \text{ with} \tag{17}$$

$$m = \frac{T}{T_{\text{cyc}}}, \tag{18}$$

where T denotes the filter time and T_{cyc} the cycle time, meaning that m denotes the size of the filter window.

To reduce the computational cost for use in online implementation it is applied as

$$M^{(m)}(t) = M^{(m)}(t-1) + \frac{x(t) - x(t\text{-}m)}{m}. \tag{19}$$

It must be ensured that, for the first calculation, $M^{(m)}(t-1)$ contains the correct starting value.

In the following an algorithm is presented, which uses n moving average filters to produce the desired position trajectory y_{f} and its six derivatives in each time step. Each filter has its own filter time T_i, with i ranging from 1 to n (here $n = 6$). As a starting point a linear point-to-point motion and its rectangular velocity are passed on to the algorithm. As part of the process different trajectories will be created, which can be distinguished by their degree of continuous differentiability. Additionally, each element of each trajectory (such as the position, velocity, …) will be described via the shape of its completed profile, even though this profile is incomplete for almost the entire filtering process. The algorithm, as described in the following, is executed once every control cycle.

Each cycle, the algorithm receives only the current values of its input trajectories, such as position and velocity. As a first step, the derivative of the trapezoidal profile is calculated. For the initial invocation in each cycle, that trapezoidal profile is the position. This leads to a rectangular shaped derivative (see Fig. 6). Next all existing elements of the trajectory are filtered using a moving average with the first filter time T_1, resulting in a C^1-trajectory and a trapezoidal velocity profile. This also leads to an increase in positioning time by T_1 (see Fig. 6). Next, the derivative is calculated once again and all elements are filtered, now with the next filter time T_2. This is repeated until it yields a C^6-trajectory. Following this method, only differentiation of trapezoidal profiles is required. Thus, a simple difference quotient can be used without creating any numerical errors. An overview of the number of derivatives and filters that have to be calculated or applied respectively, can be found in Tab. II.

For experimental validation, this algorithm is used with one adjustment. A pre-existing trajectory planning software is used, which creates C^2 -trajectories and uses the results as starting point for the filtering algorithm. This has the advantage of simplifying the implementation and making use of an existing user interface. It also reduces the number of required calculations, as listed in Tab. II. In the final implementation, computational cost is further reduced by calculating the inverse of quotients and replacing a majority of divisions with multiplications. Even without a more detailed analysis of computational costs, it becomes apparent, that the online capability of this method extends to systems with an order above $n = 6$.

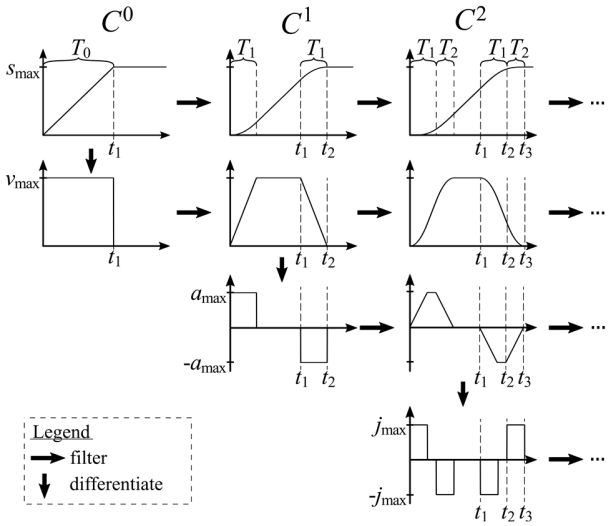

Figure 6: Procedure for the online generation of Cn-trajectories

TABLE II: COMPUTATIONAL COST OF FILTERING ALGORITHM

	derivative	**filter**
$\mathbf{C^0 \rightarrow C^6}$	6	27
$\mathbf{C^2 \rightarrow C^6}$	4	22
$\mathbf{C^k \rightarrow C^n}$	$n-k$	$\sum_{i=2+k}^{n+1} i$
number of sums	1	2
number of divisions	1	1

In section IV it will be shown, that this approach can be successfully used on the test bed, which is a real-time system. Therefore, it solves the problem of online generation of dynamic trajectories for a flatness-based control pertaining to systems with a similar system order.

C. Meeting the boundary conditions

One important aspect of trajectory planning has so far been ignored. This is the meeting of boundary conditions. It is mandatory that an adherence to a maximum velocity, acceleration and force, from now on referred to as system maxima, can be guaranteed before the start of the motion. Approximations for all system maxima can be created using only the maxima of the desired trajectory $y^{(i)}_{\text{f,max}}$. These maxima can be calculated from the filter times and preceding maxima as presented by Beckmann, et. al [8]:

$$y_{\mathrm{f,max}}^{(i)} = \frac{y_{\mathrm{f,max}}^{(i-1)}}{T_i}, \tag{20}$$

where $y_{\mathrm{f,max}}^{(i)}$ denotes the maxima of the desired trajectories i^{th} derivative and T_i denotes the matching filter time. This calculation only works for all maxima, if each filter time is equal or larger than all following filter times:

$$T_i \geq \sum_{j=i+1}^{n} \mathrm{T}_j. \tag{21}$$

If this condition is violated larger values than the calculated maxima can occur, which could lead to a violation of the system maxima.

Assuming that distance $y_{\mathrm{f,max}}^{(0)}$, maximum velocity $y_{\mathrm{f,max}}^{(1)}$ and acceleration $y_{\mathrm{f,max}}^{(2)}$ are already determined by the C^2-starting-trajectory, the corresponding filter times T_1 and T_2 can be calculated based on (20). These times do not actually correspond to a filter, since their effect was already applied in the preceding motion planning, as depicted in Fig. 6.

The remaining filter times can be chosen freely within the boundaries set by (20) and (21). Since they directly influence the system maxima, the choice of filter times influences the behaviour of the overall flatness-based control.

Following the algorithm presented in this and the previous section, an appropriate C^6-trajectory can be generated online while adhering to the given boundary conditions. The resulting trajectory y_{f} is inserted into (13), (14) and (15) to calculate feedforward term and reference trajectory respectively. This completes the implementation of the suggested flatness-based feedforward control approach.

IV. Experimental Results

In order to validate the developed flatness-based feedforward control (from now on called flatFFC) it is compared with two input shaper variations. An input shaper, also known as a command shaper, adjusts a reference trajectory in order to eliminate oscillations of a specific frequency [16]. Mathematically speaking, it convolves the trajectory with two or more impulses, as shown in Fig. 7.

By applying two impulse with the right amplitudes, at the right time, one specific frequency of oscillation can be eliminated from the trajectory and thus, from the system response. In the following, a zero vibration (ZV) input shaper

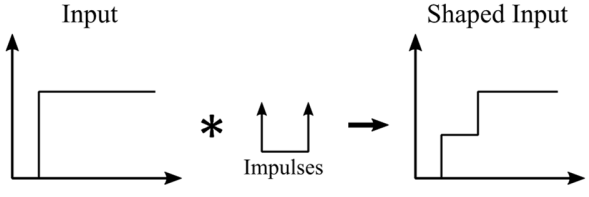

Figure 7: Basic principle of an input shaper

and an extra insensitive (EI) input shaper will be used. The ZV input shaper is the most basic variation. It uses two impulses as shown in Fig. 7 to suppress one specified frequency. In contrast the EI input shaper uses three impulses to suppress two frequencies, one slightly larger and one slightly smaller than the specified frequency. The normalised oscillation amplitude remaining after the use of these input shapers is presented in Fig. 8.

These two input shaper variations have already been used by Post et al. [4] as a comparison for a flatness-based feedforward control developed for a tower crane.

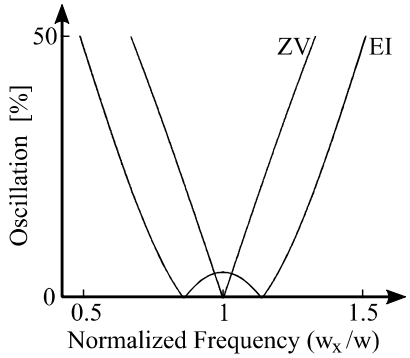

Figure 8: Sensitivity Curves of the two Input Shapers

In order to measure the mast oscillations, an acceleration sensor is attached to the mast head. The resulting measurements are processed with a low-pass filter, eliminating all frequencies above 8 Hz. The resulting measurement is structured in four parts (see Fig. 9). First the acceleration phase, next a motion with constant velocity followed by the deceleration phase and finally the oscillation phase. The amplitude of the resulting acceleration oscillation A_{a} can be converted into the amplitude of the position oscillation A_{x}, if oscillation frequency f is known:

$$A_{\mathrm{x}} = (2\pi f)^2 A_{\mathrm{a}}. \tag{22}$$

The mast is defined as oscillation free, when the oscillation amplitude no longer exceeds 2 mm. A corresponding acceleration boundary, calculated based on the observed oscillation frequency, is included in each plot. For evaluation purposes the positioning time t_{p}, oscillation time t_{o} and "oscillation free" time t_{of} are calculated for each measurement. The "oscillation free" time refers to the time from the start of the positioning, until the status oscillation free is reached.

In order to test functionality of all three methods, a trajectory with boundary conditions corresponding to Tab. III is applied repeatedly to the test bed. The same cascade control is used as a follow-up control for all methods.

For the first motion the LHD is at rest at the bottom of the mast ($h_{\mathrm{LHD}} = 0$ m). For the second motion it is at rest at the top of the mast ($h_{\mathrm{LHD}} = 1.1$ m). For the final one it moves

TABLE III: BOUNDARY CONDITIONS OF THE REFERENCE TRAJECTORY

x_{start}	x_{end}	v_{max}	a_{max}
0.3 m	2.5 m	1.9 m/s	5 m/s^2

alongside the platform from 0 m to 1.1 m. For the first two measurements the input shapers are parameterised with the same frequency as the flatness-based control, i.e. the actual mast frequency. For the final one the medium mast frequency is used to parameterize the input shapers.

As can be seen in the upper part of Fig. 9, all methods work well if the LHD is located at the bottom of the mast. While EI shows larger oscillations than the other two methods, it still barely exceeds the oscillation boundary. Generally speaking, a resulting oscillation time under half a second, as observed here, can be classified as negligible. A poorer performance of EI is expected in this case, since the input shapers are parameterised with a close approximation of the mast frequency. Therefore, the normalized frequency is approximately 1, which causes some oscillations to remain for EI (see Fig. 8).

This becomes more apparent, if the LHD is positioned at the top of the mast (see Fig. 9, middle plot). Here EI performs noticeably worse. Due to the higher position of the LHD, the effective mast mass increases to twice its size (see Fig. 5). Therefore, the resulting force on the mast head doubles as well. This results in a remaining oscillation of around 5 mm for EI. Due to the minimal damping of the mast this increases the "oscillation free" time t_{of} significantly. The resulting characteristic times as defined beforehand are listed in Tab. IV.

An important difference between the three methods is the positioning time t_p. ZV has the shortest t_p out of the three methods. EI has a longer positioning time due to its convulsion with three impulses instead of two. The positioning time of flatFFC is slightly longer than that of ZV. Since, for the first two scenarios, both ZV and flatFFC show no noteworthy oscillations following the positioning, this means that ZV has the best overall results for a motion with a static LHD.

It should be noted that each method increases the positioning time compared to the initial trapezoidal

TABLE IV: CHARACTERISTIC TIMES (BEST TIMES IN BOLD)

h_{LHD} [m]		t_p [s]	t_o [s]	t_{of} [s]
0	ZV	**1.72**	**0**	**1.72**
	EI	1.86	0.48	2.33
	flatFFC	1.77	**0**	1.77
1.1	ZV	**1.79**	**0**	**1.79**
	EI	1.91	6.31	8.22
	flatFFC	1.80	**0**	1.80
0 – 1.1	ZV	**1.74**	6.30	8.20
	EI	1.89	2.56	4.34
	flatFFC	1.78	**0.13**	**1.87**

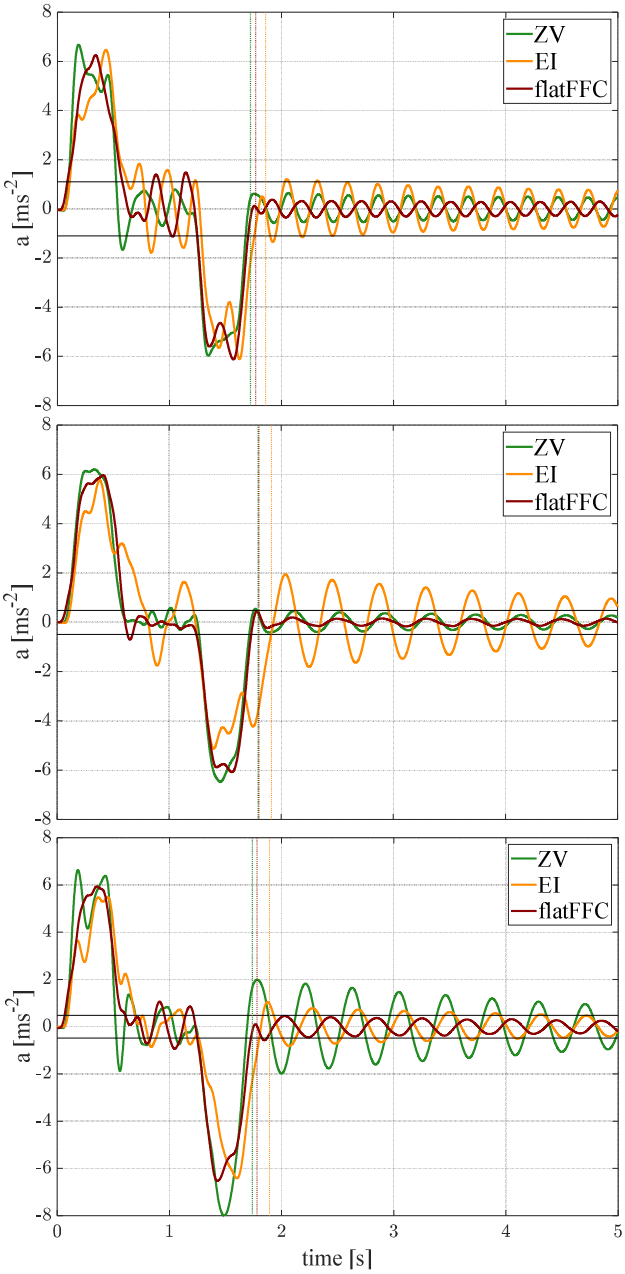

Figure 9: Resulting mast acceleration for three different positioning tasks. For each method, the end of the positioning is marked by a vertical line. (top: h_{LHD} = 0 m, middle: h_{LHD} = 1.1 m, bottom: h_{LHD} = 0 m – 1.1 m)

acceleration profile (t_p = 1.64 s). This is compensated by a reduced oscillation time t_o resulting in an overall lower t_{of}. For instance, for the first measurement (see Fig. 9, top) the same motion without using an oscillation reduction scheme results in a t_{of} of 8.42 s. In [6] and [8] where a similar trajectory generation scheme to flatFFC is used, such an increase in positioning time is avoided. The jerk of the flatness-based trajectory is increased, so that it compensates the lengthening in positioning time caused by the trajectory

filtering. This is more difficult in this work, since a C^6-trajectory is required in contrast to the C^4-trajectory used in [6], [8]. A higher order of differentiability entails more filtering and thus, a larger increase in positioning time.

Regarding the oscillation reduction, similar results are achieved. In [6] and [8] the maximum reduction of oscillation amplitude is 89%. Applying the same criteria here, leads to a reduction of 94% for flatFFC, and 72% and 38% for ZV and EI respectively.

For a simultaneous motion of platform and LHD the effectiveness of the two input shaper types inverts, leading to a better result for EI. ZV is not capable of effective damping, if the eigenfrequency changes during platform motion. In contrast, EI is designed to dampen a broader spectrum of frequencies. Therefore, it performs better for a moving LHD than for a static one. Even better performs flatFFC with a negligible oscillation time. This proves that the developed oscillation reduction scheme is able to drastically reduce oscillations for a moving LHD as well.

With regard to literature, only the results of [3] and [4] can be compared to the simultaneous motion of platform and LHD. [3] achieves a completely oscillation free positioning. However, further comparison is difficult, since the paper employs an optimization-based trajectory planning approach. In [4] the flatness-based approach is compared to EI and ZV input shapers as well, but it is applied to a tower crane. Yet it also examines a simultaneous motion of two axis, comparable to simultaneous motion of platform and LHD. The results regarding positioning time and oscillation reduction exhibit the same trends as the validation of the simultaneous motion presented here. Meaning it finds the flatness-based approach to offer the best compromise regarding both criteria, with near optimal oscillation reduction but a slightly longer positioning time.

Overall it can be stated that while some oscillations remain during simultaneous motion, the chosen modelling approach that includes the time-variant mast eigenfrequency has proven an effective basis for a flatness-based oscillation reduction scheme.

A substantial negative effect due to neglection of the upper mast eigenfrequencies cannot be observed. The only case that leads to any remaining oscillation is the simultaneous motion. Yet this case is not examined in works which include these neglected frequencies ([2], [5], [10]), making further analysis difficult.

Regarding input shapers, it was shown that EI offers a variation of input shaping that can reduce oscillations for a moving LHD, yet only flatFFC is able to almost completely eliminate them. Since for a real stacker crane most motion would be simultaneous motion of LHD and platform this last result is of the highest importance. It should be noted that, in real life applications, not all LHD motion would span the entire mast length, thus possibly causing a smaller change in mast eigenfrequency.

V. Conclusion

In this paper, a flatness-based feedforward control approach for oscillation reduction on a belt driven stacker crane has been presented. A 6th order model of the test bed is created, taking into account belt and mast oscillations as well as the time-variant nature of both eigenfrequencies. The inverse system model necessary for feedforward control is calculated using the method of differential flatness.

A new method for generating desired trajectory is introduced, which uses moving average filters to increase the continuous differentiability of a s-curve trajectory with a trapezoidal acceleration profile. By using a causal filter, the current values of the desired trajectory can be calculated in each control cycle. This distributes computational costs across the motion, reducing the computations necessary in one cycle. The adherence to boundary conditions can be guaranteed by choosing appropriate filter times. Through this method online generation of sufficiently smooth, time efficient trajectories on a real-time system becomes possible. This in turn enables the use of flatness-based feedforward control without requiring lengthy offline calculations beforehand and without foreknowledge of the desired trajectory. Only an initial parametrization of the model is required. Additionally, it can easily be implemented on a variety of systems, using a variety of control schemes, by simply inserting the feedforward control after the already existing motion planning.

The developed feedforward control scheme proves capable of reducing the mast oscillation to negligible levels. This is achieved with a much simpler dynamic model than many other attempts at flatness-based feedforward control of stacker cranes use. It has worked consistently well for all tested scenarios, matching or exceeding the performance of the input shaping. When the load handling device is moved during motion, a 95 % reduction of oscillation time compared with the best input shaper is achieved. The consistent performance of this oscillation reduction scheme for a static and a moving load handing device sets it apart. This is primarily enabled through the inclusion of the time-variant mast eigenfrequency in the system model.

References

[1] J. Bond, „Modern Materials Handling," 2018. [Online]. Available: shorturl.at/etCWY.

[2] M. Bachmayer, Effizientes Positionieren am Beispiel Regalbediengerät, Technische Universität München, 2012.

[3] M. Staudecker, K. Schlacher und R. Hansl, „Passivity Based Control and Time Optimal Trajectory Planning of a Single Mast Stacker Crane," in *The International Federation of Automatic Control*, Seoul, Korea, 2008.

[4] B. K. Post, A. Mariuzza, W. J. Book und W. Singhose, „Flatness-based Control of Flexible Motion Systems," in *Dynamic Systems and Control Conference*, Arlington, VA, USA, 2011.

[5] M. Bachmayer, H. Ulbrich und J. Rudolph, „Flatness-based control of a horizontally moving errected beam with a pointed mass," in *Mathematical and Computer*, 2011.

[6] D. Backmann, Online-Parameteridentifikation mechatronischer Antriebsstränge zur aktiven Schwingungsdämpfung, TEWISS Verlag, 2018.

[7] J. Levine, Analysis and control of nonlinear systems: A flatness-based approach, Springer Science & Business Media, 2009.

[8] D. Beckmann, M. Schappler, M. Dagen und T. Ortmaier, „New Approach using Flatness-based Control in High Speed Positioning: Experimental Results," IEEE, Leibniz Universität Hannover, 2015.

[9] D. I. B. Berger, „Method for controlling a storage and retrieval unit". Austria Patent AT514152B1, 08 April 2013.

[10] M. Schipplick und W. A. Günthner, „Potentiale einer flachheitsbasierten Antriebssteuerung am Beispiel Regalbediengerät," 2011.

[11] J. E. Pope, Rules of Thumb for Mechanical Engineers, Elsevier, 1996.

[12] M. Paz, in *Structural Dynamics: Theory and Computation*, New York, Chapman & Hal, 1997, pp. 133-134.

[13] W. Bopp, Untersuchung der statischen und dynamischen Positionsgenauigkeit von Einmast-Regalbediengeräten., Karlsruhe: Dissertation, 1993.

[14] M. Fliess, J. Levine, P. Martin und P. Rouchon, „On differentially flat nonlinear systems," in *Nonlinear Control Systems Design 1992*, 1993, pp. 159-163.

[15] P. Lambrechts, M. Boerlage und M. Steinbuch, „Trajectory planning and feedforward design for electromechanical motion systems," in *Control Engineering Practice 13*, 2005, pp. 145-157.

[16] T. Singh und W. Singhose, „Tutorial on Input Shaping/Time Delay Control," *Proceedings of the 2002 American Control Conference,* Nr. Vol. 3, p. 16, 2002.

[17] M. S. A. Braun, Entwicklung, Analyse und Evaluation von Modellen zur Ermittlung des Energiebedarfs von Regalbediengeräten, Karlsruhe, 2016.