PLC-Based Real-Time Realization of Flatness-Based Feedforward Control for Industrial Compression Systems

Shane Dominic, Yannik Löhr, Andreas Schwung, and Steven X. Ding

***Abstract*—In this paper, we present a novel programmable logic controller (PLC)-based real-time realization of a flatness-based feedforward control (FFC) scheme. The proposed approach is applied to an industrial fuel-gas compression system which is used to supply fuel gas to the gas turbines in combined cycle power plants. Due to the increasing demand for fast operation point transitions with high performance and accuracy requirements, the currently applied decentralized proportional-integral-derivative controllers appear to be not appropriate any more. Hence, by means of system simulations, a new flatness-based FFC design has been shown to provide improved control performance. In this paper, we bridge the gap between simulation-based control design and practical applicability, in that, we present the real-time realization of the approach on a PLC. Furthermore, the PLC-based controller is tested on a hardware-in-the-loop platform running with a complex compression system model in real time. The results reveal that the flatness-based control design can be implemented on a real compressor system.**

***Index Terms*—Advanced control, flatness-based approach, fuel-gas compression process, real-time implementation.**

I. Introduction

MANY control concepts show great potential to provide solutions for industrial problems. Studies have shown that, despite numerous research and development of advanced control concepts in the process industry, up to 95% of all control loops are still PI(D) type controllers [1]. The reasons are low computational cost and good performance, due to the small number of parameters, which have to be tuned [2]. However, the increasing demand for high-performance applications and higher energy efficiency requires for more sophisticated controller design.

Manuscript received March 8, 2016; revised May 24, 2016; accepted July 25, 2016. Date of publication September 21, 2016; date of current version January 10, 2017.

S. Dominic and S. X. Ding are with the University of Duisburg-Essen, 47057 Duisburg, Germany (e-mail: shane.dominic@uni-due.de; steven.ding@uni-due.de).

Y. Löhr is with the Ruhr-University Bochum, 44801 Bochum, Germany (e-mail: yannik.loehr@rub.de).

A. Schwung is with the South Westfalia University of Applied Sciences, 59494 Soest, Germany (e-mail: schwung.andreas@fh-swf.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2016.2612160

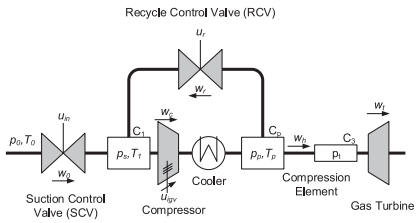

Fig. 1. Installation of a fuel-gas compressor system.

One example is compression systems which are used for a plenty of different industrial applications such as air separation units, blast furnaces, fluid catalytic cracking units, and fuel-gas compression packages. In the latter case, the compressor is used to supply fuel gas to the gas turbines (GTs) in combined cycle power plants (CCPP) which still play a major role in the energy management of modern energy networks. Their short start-up times and abilities for fast load changes make CCPP a promising alternative for mid-to-peak-load power plants, especially as a complement technology to regenerative energy sources. However, requirements for continuous uninterrupted operation as well as the ability for fast operation point changes poses high challenges to the fuel compression system. To ensure a continuous supply of fuel gas, highly precise controllers for the compression system are required. Especially, fast adaptation capability to the actual load condition in the network has to be managed by the control system. A schematic of the system is shown in Fig. 1.

To date, the focus of the control design for compression systems is on the compressor instabilities causes by surge or stall [3], [4]. Based on the well-established Moore-Greitzer compression model [5], [6], which describes the basic dynamics of the surge phenomenon, control approaches are designed such as a linear control approach [7], nonlinear controller [8]–[10], Lyapunov-based methods [11], [12], adaptive approaches [13], and linear parameter varying controller [14].

However, for the considered fuel-gas compressors, not only the surge avoidance but especially the operation within the operating range during fast load transitions and a fast start-up of the system are the main control objectives. Currently, three decentralized proportional-integral-derivative (PID) controllers are used for this purpose, which control the suction pressure, the discharge pressure, and the compressor volume flow. However, these controllers show a poor performance, when fast oper-

ating point transitions are required [15], [16] and are, therefore, not appropriate for an industrial application. In [15], a new advanced control approach is presented, using a flatness-based feedforward control (FFC) approach with online computation of feedforward trajectories [17]. The differential flatness property of a system originally presented in [18] has successfully been employed in different applications like power electronics [19]–[22], distributed power generators [23], and high-speed linear axis [24] to name a few. Also for the considered compression system, the flatness-based approach has shown its great potential also for improved control during simulations. However, the control code implemented in MATLAB/Simulink is not applicable for real-time operation.

During the control design, new control methods are developed in order to improve the control performance. Using monitoring and detection techniques, the control performance can be also improved [25], [26]. But normally, less consideration are made about the implementation on a real control unit. Especially, the real-time capability and the required computation power of the algorithms are usually not focused while designing the controller. The results of a normal PC simulation are normally hard to be realized in the control unit such as programmable logic controller (PLC). In this paper, we bridge this gap in that we present a PLC-based real-time capable realization of the flatness-based FFC with online trajectory generation. This enables the application of the control scheme for real-world compression system models. Thereby, the previously applied PID controller can be integrated in the presented framework facilitating the industrial implementation. Furthermore, we present results of the practical verification of the implemented control strategy on a hardware-in-the-loop (HIL) platform. On this HIL platform, the PLC is connected to a detailed and precise compression system model [27], which can be simulated in real time. In the literature, most common PLC-based controller have the standard PID form [28]. In [29], a method for improving the robustness of PID control using a two-loop model following control system is presented. In [30], a multivariable hysteresis type controller for induction motors is proposed. However, very little approaches have been presented for the implementation of advanced control algorithms on PLC-based systems. In [31], fuzzy-logic-based controllers are implemented in a PLC system. Giannoutsos and Manias [32] present a PLC-based self-tuning data-driven controller. An application of a PLC-based control to reciprocating compressors is shown in [33]. However, to the author's knowledge, this is the first approach for a real-time PLC-based implementation of flatness-based control systems.

II. System Description and Problem Formulation

The challenges we consider in this paper are fast operating point changes of industrial compression systems as they typically appear during fast compressor start-up and fast and sudden load changes of the subsequent process. The considered application example is a fuel-gas compressor installation which supplies fuel gas from the header pipeline to the combustion process of the GT. To allow for continuous operation of the GT, the mass flow, w_h, as well as the inlet pressure of the turbine, p_t, has to be controlled within tight limits especially during fast load changes. Fig. 1 shows the complete system.

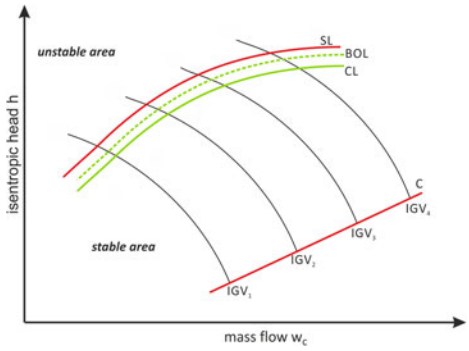

Fig. 2. Normalized compressor map.

The system installation consist of the compressor, a suction control valve (SCV), a cooler and a recycle control valve (RCV). The compressor is powered by an electrical motor which keeps the operating speed constant. The pipeline provides gas with a certain mass flow, w_0, and temperature, T_0. By means of the SCV, the suction pressure, p_s, can be controlled up to maximum allowable inlet pressure, $p_{s,\max}$. The gas is then compressed to the discharge pressure, p_d. To cool the compressed gas from T_c to the desired inlet temperature, T_d, of the turbine a cooler is integrated. To regulate the inlet turbine mass flow w_h and to protect the compressor against surge, the RCV is used. By opening the valve the compressor mass flow w_c can be reduced to $w_h = w_c - w_r$. When the operating point crosses the control line, an antisurge controller gradually opens the RCV. When crossing the blow-off line (BOL), the valve is fully opened.

The operating range of the fuel-gas compressor is characterized by the nonlinear characteristic map, as shown in Fig. 2. The normalized map contains the specific enthalpy difference Δh as a function of the mass flow, w_c, and the compressor inlet guide vanes (IGV) position. For small w_c, the compressor's operating region is constrained by the surge line (SL); for large w_c by the choke (C). The SL divides the compressor map into stable and unstable operating points. By crossing the SL, high mechanical loads can be induced and can cause damages to the compressor blades. Thus, an operating of the compressor to the left of the SL has to be avoided by opening the RCV. On the other hand, if w_h reaches the velocity of sound, the operating point will cross the choke.

As already mentioned, the system is controlled by three decentralized PI controllers where the IGV position and the position of RCV and SCV are used as actuators, with actuating variables, u_{igv}, u_r, and u_{in}. Since the motor speed is constant, the whole operating range is controlled by the IGV position. Changing u_{igv}, the discharge pressure can be regulated while the inlet pressure is controlled by u_{in}. For compressor protection and surge control, u_r is used. To avoid interactions of the controllers during operation, a distinction between process control and surge control is made. In an event of fault or disturbance, surge controller has got a higher priority. More detailed information about the fuel-gas compression process is given in [15] and [16].

For the process, mainly three different types of operating point changes can occur: startup and shutdown, operating point changes, and load change. The load change is the most demanding one as a reduction up to 25%–30% of the nominal turbine

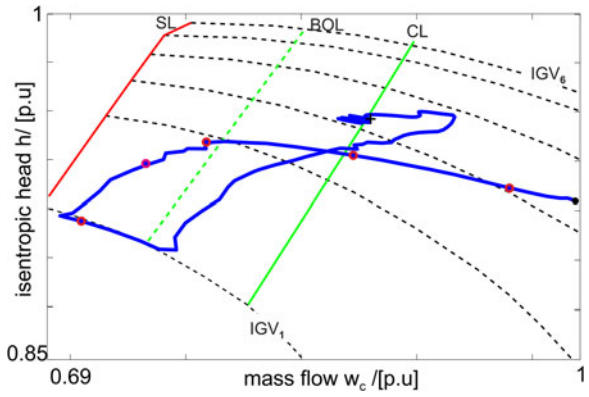

Fig. 3. Simulation of a load rejection with the current PID controller.

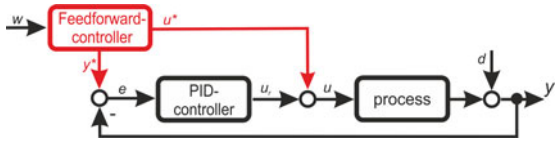

Fig. 4. Two-degree-of-freedom control loop.

mass flow within 1 s has to be considered, which is also called load rejection. This abrupt change can lead to an undesired and critical performance of the control system especially as the process constraints must be respected at all times. In addition, the GT manufacturer specification allows a maximum deviation of the turbine inlet pressure by 5% with additional constraints like the rate of the pressure change or the pressure amplitude.

Fig. 3 illustrates such a load rejection example and the performance of the implemented decentralized controller. It clearly demonstrates the poor behavior of the control concept. The blue points with the red border indicate the first 5 s after load rejection occurs. Each point indicates 1 s. The operating point crosses BOL after 3 s. This leads to a full opening of the RCV and a high-discharge pressure drop, which violates the given constraints of the turbine and causes a trip of the whole plant. This clearly indicates that the current control scheme is not suitable for such operating conditions. Therefore, it requires a redesign of the existing control structure.

The first two aforementioned operating point change conditions are not that critical from the control system stability point of view due to the comparably small slope of the transients. However, particularly for parallel operation scenarios a fast start-up of a stand-by compressor is required. Also for such scenarios the flatness-based FFC presented next is particularly suitable.

III. Flatness-Based FFC Approach

In [15], a novel two-degree-of-freedom control scheme for the fuel-gas compressor system has been presented. The derivation of this approach is summarized here. The new structure, as shown in Fig. 4, contains the already existing PID feedback controller and an additional feedforward controller. Thus, the control input u is composed by both control input u^* and u_c.

The design of the feedforward controller is model-based and determines the control input u^* only by the given new set point w. To compensate for model-mismatch and disturbances d, the PID feedback controller is used, which can be seen as a second degree of freedom. Based on the difference between the desired output trajectory y^* and the plant output y, the PID controller adds an additional value u_c to u in order to reduce the error e. Hence, the existing PID controller can be preserved under the new controller structure.

The idea of the new control approach is to regulate the system from the actual operating point y to the given new set point w by evaluating proper trajectories u^* and y^*. Designing the feedforward controller based on the differential flatness property, u^* and y^* can be computed analytically without solving differential equations in real time. In the literature, two different kinds of FFC design exist: offline [34] and online [35]. Due to the limitation of an offline design, an online design is developed in [15], in order to treat a higher number of operating point transitions. This concept can be seen as a high potential "two-degree of freedom control design tool," which can be implemented on almost all types of flat systems.

A. Definition of Flatness

In 1992, Fliess *et al.* introduced a new concept of differential flatness [18], which was developed for single-input single-output nonlinear systems and, then, extended to multi-input multi-output nonlinear systems [36]. Flat systems can be seen to a certain extent as a simplification of nonlinear systems. In a suitable coordinate system, a nonlinear flat system behaves similarly to a linear system in a linear coordinate system. Thus, flat systems expand the concept of controllability of nonlinear systems, and hence, the possibility of trajectory planning is given. We consider the nonlinear dynamic system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \mathbf{x}(\mathbf{0}) \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m. \tag{1}$$

Then, the flat output $\mathbf{y}_f \in \mathbb{R}^m$ can be written as

$$\mathbf{y}_f = \boldsymbol{\psi}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \ldots, \mathbf{u}^{(\beta)}), \tag{2}$$

with the state $\mathbf{x}$, the input $\mathbf{u}$, and a finite number of input derivatives with respect to time. Using $\mathbf{y}_f$ and a certain number of time derivative, $\mathbf{x}$ and $\mathbf{u}$ can be calculated as

$$\begin{aligned} \mathbf{x} &= \boldsymbol{\psi}_1(\mathbf{y}_f, \dot{\mathbf{y}}_f, \ldots, \mathbf{y}_f^{(\beta)}), \\ \mathbf{u} &= \boldsymbol{\psi}_2(\mathbf{y}_f, \dot{\mathbf{y}}_f, \ldots, \mathbf{y}_f^{(\beta+1)}). \end{aligned} \tag{3}$$

The important feature is that all system and input variables can be expressed with algebraic relations as a function of the flat output $\mathbf{y}_f$ and its time derivatives. Hence, the trajectory of system and input variables can be directly determined from a given trajectory of the flat output. Thereby, no complex differential equations have to be solved, which is a benefit for a real-time realization, due to the smaller computation power of control units like PLCs.

B. Design of the Spline Trajectory

If an operating point change happens from $y(t_0)$ to $y(t_0 + T)$, then the desired trajectory for the flat output $\mathbf{y}_f$ has to be designed, in order to bring the system from the current operating condition $\mathbf{y}_{f,0}$ to the final one $\mathbf{y}_{f,T}$ within the given time T. As

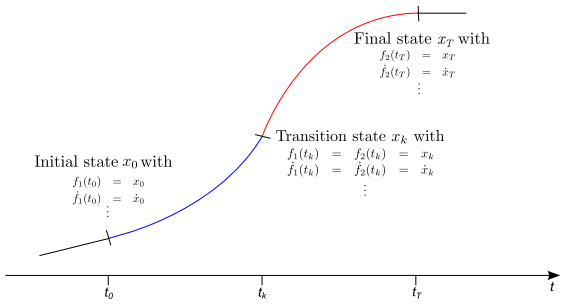

Fig. 5. Typical spline-based trajectory.

shown in Fig. 5, for each component of $\mathbf{y}_f$, splines can be defined by two polynomial functions f_{1i}, f_{2i} $(i = 1 \ldots, m)$ of order η as

$$f_{1i}(t) = \sum_{i=0}^{\eta} d_{1i} t^i, t \in [t_0, t_k],$$

$$f_{2i}(t) = \sum_{i=0}^{\eta} d_{2i} t^i, t \in [t_k, t_T]. \tag{4}$$

Hence, for $t_0 \leq t \leq t_k$, $\mathbf{y}_f$ is set to $\mathbf{y}_f(t) = \mathbf{f}_1(t)$ and for $t_k \leq t \leq t_T$, $\mathbf{y}_f(t) = \mathbf{f}_2(t)$. Following (4), the trajectories have to be at least $\beta_i + 1$-times continuously differentiable and, therefore, the order η has to be defined as $\eta \geq \beta_i + 1$.

To determine the unknown parameter d_{1i}, d_{2i} the given boundary conditions are used. These constraints include the initial, transition, and final state of $\mathbf{y}_f$ and also the requirement for continuity and smoothness of the polynomial functions. These can be set using three conditions,

$$f_1^{(l)}(t_0) = y_{f,0}^{(l)}, f_1^{(l)}(t_k) = f_2^{(l)}(t_k), f_2^{(l)}(t_T) = y_{f,T}^{(l)}, \tag{5}$$

with $l = 0, \ldots, \beta + 1$. This yields a linear system of equations with order $N = 3(\beta + 1)$, which has to be solved in order to calculate d_{1i} and d_{2i}. Since, there are highly optimized algorithms for the solution of linear systems, this is another benefit for the real-time realization.

IV. Modelling of the Fuel-Gas Compressor System

To design the feedforward controller, a model of the compressor system is developed based on [27]. Since the model is derived in [15], just a summary of the results is presented here. Assumptions to simplify the modeling consist among others in the examination of the medium as an ideal gas and a constant density. A further simplification is the merging of all pipelines into three volume elements, suction and discharge volume V_s, V_d of the compressor, and the inlet GT volume V_t. The pressure in these volumes can be obtained by differential equations as a function of all incoming and outgoing mass flows and their respective heat capacities,

$$\dot{p}_s = \frac{1}{C_s}(w_0 + w_r - w_c),$$

$$\dot{p}_d = \frac{1}{C_d}(w_c - w_r - w_h),$$

$$\dot{p}_t = \frac{1}{C_t}(w_h - w_t). \tag{6}$$

The mass flows through the compressor, RCV and SCV are described by w_c, w_r, and w_0. The compressor outlet mass flow and the GT inlet mass flow are expressed by w_h and w_t. The heat capacity can be written as

$$C_i = \frac{c_i^2(T_i)}{V_i}, \tag{7}$$

with $i \in [s, d, t]$ and the velocity of sound c_i,

$$c_i = \sqrt{\kappa R T_i}, \tag{8}$$

with the gas constant R, the isentropic exponent κ, and the temperature T_i of each of the three elements. The mass flows in (6) can be described as

$$\dot{w}_c = \frac{A_c}{l_s}(p_s - p_d), \quad \dot{w}_h = \frac{A_t}{l_t}(p_d - p_t), \tag{9}$$

where l is the pipe length and A is the cross-sectional area of the respective element. Using a simplified valve equation, the other two mass flows are

$$w_0 = k_{v,\text{in}} u_{\text{in}} \sqrt{p_0 - p_1}\,\text{sgn}(p_0 - p_1),$$

$$w_r = k_{v,r} u_r \sqrt{p_p - p_1}\,\text{sgn}(p_p - p_1). \tag{10}$$

with sgn as the sign function. The input values of the valve position for RCV and SCV are given by u_r and u_{in}. The valve specific flow coefficients are described by $k_{v,r}$ and $k_{v,\text{in}}$. Furthermore, p_0 and p_d denote the pressure of the supply pipeline, which is a measurable disturbance, and the compressor discharge pressure, which can be obtained by the following relationship:

$$p_d = \Phi(h(w_c, u_{\text{igv}}), T_s) \cdot p_s. \tag{11}$$

The compressor characteristic is described by Φ, which is a function of the compressor head h and the suction temperature T_s. As it is shown in Fig. 2, h is a function of the mass flow w_c and the IGV position u_{igv}. Since, the temperatures T_d and T_0 are given, only the discharge temperature has to be calculated, as follows:

$$T_s = \frac{T_0 w_0 + T_p w_r}{w_0 + w_r}. \tag{12}$$

Using (6)–(12), the nonlinear state-space model can be formulated with the input vector $\mathbf{u} = (u_{\text{igv}}, u_r, u_{\text{in}})$ and state vector $\mathbf{x} = (p_s, w_c, p_d, w_h, p_t)$. The external disturbances are summarized to the vector $\mathbf{d} = (p_0, T_0, w_t)$.

The approximation of the compressor map, as shown in Fig. 2, represents a major challenge for modeling and implementation. Based on the nonlinear characteristic, the feedforward trajectory for u_{igv} has to be calculated. The map is normally given as a lookup table. However, an online interpolation via lookup table is impossible due to the low computational power of control units. Hence, an analytical relationship for w_c, u_{igv}, and h is required for the feedforward controller design where the nonlinear compressor characteristics have to be modeled as accurately as possible. Unfortunately, the Moore-Greitzer compressor model shows too large deviation from the true compressor map. Thus, a new approximation is required [15]

$$u_{\text{igv}} = a_1 \arctan\left(a_2 + a_3 h + a_4 w_c + a_5 \frac{h}{w_c}\right) + a_6. \tag{13}$$

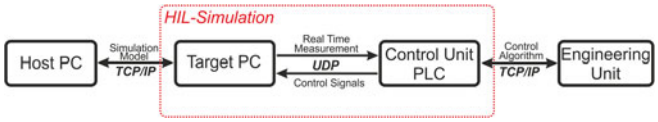

Fig. 6. Scheme of the HIL-Platform.

The variables $a_i, i = 1, \ldots, 6$ are fitted to the data of the lookup table. With (13), an approximation error less than 5% can be achieved.

In [15], it is shown that the suitable flat output for the considered system is,

$$\mathbf{y}_f = [p_s \; w_s \; p_t]^T , \tag{14}$$

which satisfy the differential flatness conditions.

V. Real-Time Realization

To evaluate the practicability of the proposed control approach, an HIL platform has been realized, as shown in Fig. 6. Based on a real-time simulation, which contains an accurate model of the process, the control performance can be analyzed and applicability for real application can be noted. Furthermore, it has to be verified, if the control unit is able to run such a complex control algorithm as developed in this approach.

A. HIL-Platform

The MATLAB toolbox xPC-Target is a powerful toolkit to run complex simulation models in real time. The simulation models can be developed on the Host PC and after compilation, the models are transferred to the Target PC via TCP/IP communication. The Target PC is used to simulate the model in real time. Thus, as the fuel-gas compressor model has a high accuracy to the real process, it will have nearly the same behavior as the real plant. The control algorithm is realized using the Siemens software PCS7, which is one of the standard industrial software for the process control system and supports the use of several programing languages. In order to cope with the demanding mathematics, the high-level-oriented language S7-SCL has been chosen. However, the PLC only calculates with an accuracy of 32bits, which means that there is a low precision of six digits and the presentable number range is $[1.4 \times 10^{-45} \; 3.8 \times 10^{38}]$. The engineering unit is used to implement the control algorithms, which are transferred to the control unit via TCP/IP. As a control unit a Soft-SPS is used, running on the Siemens IPC427C with a clock rate of 1.2 GHz. In HIL, the Target PC will run the model in real time and the controller will react based on the output of the model. Therefore, with the real-time measurements, the implemented controller will sent the control signals via UDP communication. The UDP protocol allows not only the communication partner to communicate in real time, but also with a small sampling time down to 2 ms. Another advantage is that the possible transfer rate is up to 100 Mb/s, which permits a high number of transfer signals.

B. Implementation of the Flatness-Based FFC

For constant operating points, the reference values are available to the PLC. Using (9)–(13), the FFC input is calculated once and kept constant until an operating point change occurs. On this account, a demand for a trajectory planning arises. As the initial value the current operating point of the compressor at time t_0 is taken. It is assumed that the final operating condition of the turbine and consequently the compressor conditions are known, which should be reached in a given time interval $t_0 \longmapsto t_T$. Note that this assumption is valid for the considered application scenarios like load rejection, known process operation transitions, and compressor start-up. By means of the initial and final conditions, the trajectory planning for the desired output $\mathbf{y}$ can be defined as $\mathbf{y}(t = t_0) \longmapsto \mathbf{y}(t = t_T)$. As the control values can be traced back to the flat output, the main task is the definition of the trajectory of $\mathbf{y}_f$. Combining (4) with the conditions (5), a linear system of equations results whose solution coefficients have to be determined in order to calculate the desired trajectory. Note that the calculation of the trajectory for the flat output $\mathbf{y}_f$ is done only at the beginning of the transition. After that, relations (9)–(13) and the real-time measurement are used to calculate the FFC input for each sampling time.

For the realization of the FFC on the PLC, we have to address the following challenges. First, an appropriate numerical method to solve a linear system of equations has to be implemented. Second, the calculation and storing of the spline function with respect to time has to be managed. Third, due to the wide numerical range of the coefficients, the evaluation of the compressor map for the trajectory generation has to be considered.

Several methods for the solution of a linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \; \mathbf{A} \in \mathbb{R}^{N \times N}, \mathbf{b} \in \mathbb{R}^N, \tag{15}$$

are presented in the literature, such as Cholesky factorization, GramSchmidt process, or SOR method [37]. In this paper, Gaussian elimination method based on LU decomposition is used. Due to the low precision of the PLC, the subtraction of two nearly equal numbers and also the division by small values has to be avoided. For this reason, a pivot matrix $\mathbf{P}$ that transforms the resulting system of equations from (5) to

$$(\mathbf{PA})\mathbf{x} = \mathbf{Pb} \tag{16}$$

is implemented in order to prevent a loss of significant digits. The objective in generating $\mathbf{P}$ is to sort the rows or columns of A by descending order of their diagonal elements. Hence, each diagonal element has to be analyzed and the considered row or column of the matrix has to be buffered. This increases the number of operations and leads to a higher memory requirement, which is however still acceptable for the control unit performance. Solving (16), the coefficients $d_{1,i}, d_{2,i}$ of (4) can be obtained. To calculate the functions f_{1i}, f_{2i} of the trajectories for $\mathbf{y_f}$, these coefficients have to be symbolically saved in the array $coeffSpl$. The following algorithm shows the implementation of the trajectory calculation on PLC. In the first step, the equation system (15) is solved by function *SolveLs* for each control value $i_m = 1$. The variable sol is used to buffer the result for each spline polynomial $i_s = 1$. Based on that, the co-

TABLE I
STORING OF COEFFICIENTS IN *Coeffspl*

id	i_m	i_β	i_s	i
0	1	0	1	1
4	2	0	1	1
8	3	0	1	1
12	1	0	2	1
16	2	0	2	1
⋮	⋮	⋮	⋮	⋮
92	3	3	2	1
96	1	0	1	2
⋮	⋮	⋮	⋮	⋮
188	3	3	2	2
192	1	0	1	3
⋮	⋮	⋮	⋮	⋮
572	3	3	2	6

efficients d of the trajectory and its derivatives $i_\beta = 0 \ldots . \beta$ are calculated by multiplication with the standard derivation of the polynomials *coeffTmp*. Finally, the values are stored in the data block *coeffSpl*, which structure is shown in Table I, where id marks the index of the used double word for each combination of i_m, i_s, i_β, and i.

The dynamic setting of (5) based on the known time instants t_0, t_k, and t_T poses an additional challenge. The system of equations includes the calculation of β derivatives of the polynomials (4). A case-by-case analysis of (5) results in the three following cases, one for each segment of the spline in Fig. 5

$$\text{line}(i) = \begin{cases} \text{case I}_{|t=t_0}, & i = 1 \ldots \frac{N}{3}, \\ \text{case II}_{|t=t_k}, & i = (\frac{N}{3}+1) \ldots (\frac{2\cdot N}{3}), \\ \text{case III}_{|t=t_T}, & i = (\frac{2\cdot N}{3}+1) \ldots N, \end{cases}$$

where Cases I and III can be computed by the same formula, allocating the first or last $k = \eta + 1$ columns of the linear system,

$$A[i,k] = \begin{cases} 0, \ (\alpha - \beta - k) \leq 0, \\ t^{(\alpha-\beta-k)} \prod_{j=1}^{\beta} ((\alpha - \beta - k) + j), \ \text{plain}. \end{cases} \quad (17)$$

The order of the coefficients are presented with α. In Case II., (16) is used twice for the computation of f_1 and f_2, which are subtracted afterward.

The remarkably low error for the approximation of the compressor map in [15] stresses the evaluation of ninth-order polynomials that include coefficients of order $b_9 = 10^{-45}$. For feasible input values, on the other hand, there are multipliers up to $h^9 = 10^{42}$. As these coefficients are not negligible but also not representable in the PLC, the implementation must include an offline adjustment of these coefficients

$$b_{\eta,\text{corr}} = b_\eta \cdot 10^{(\eta-1)\cdot q}. \quad (18)$$

Furthermore, to avoid overflow, this involves the reduction of the exponents in the online multiplications by solving the mini-

Algorithm 1: Calculate and Store Trajectories.

```
for i_m = 1 to m by 1 do
    SolveLs.DBX(A_{i_m} = LS.elements, b = con_{i_m});
                result = DBX.x;
    for i_s = 1 to s by 1 do
        for j = 1 to (η + 1) by 1 do
            sol[j] = result[(i_s − 1) · 6 + j]
        end for
        for i_β + 1 = 1 to β by 1 do
            for j = 1 to (η + 1) by 1 do do
                d[(i_β + 1), j] = sol[j] ·coeffTmp[(i_β + 1), j]
            end for
            for i = 1 to (η + 1) by 1 do do
                coeffSpl [i, i_β, i_s, i_m] = d[(i_β + 1), j]
            end for
        end for
    end for
end for
```

mization problem

$$\begin{aligned} &\min q \\ &\text{s.t.} \\ &\hat{h} = h \times 10^{-q}, \\ &(\hat{h})^{\eta-1} \leq 3.4 \times 10^{38}, \\ &q \in [1, (\eta - 1)]. \end{aligned} \quad (19)$$

After these adjustments, the polynomials can be evaluated by the given control structures of S7-SCL, despite the numerical limitations of the PLC. It is necessary to emphasize that the sequence of operations is very important during these calculations. For instance, this means that the power function always has to be computed separately and also (13) has to be split up, in order to reach a sufficient numerical accuracy.

In addition, the evaluation of the polynomials also occurs in the calculation of the trajectories of $\mathbf{y}_f$. Given that it is not possible to use symbolic addressing to read the coefficients out of a data block, the relationship

$$id = 4((i_m - 1) + m((i_s - 1) + s(i_\beta + i(\beta + 1)))) \quad (20)$$

was found in order to address the index of the ith coefficient of each feasible trajectory indirectly. The steps are summarized in Algorithm 2.

According to this parameterization, the trajectories of $\mathbf{y}_f$ within the transition are well defined and can be used to calculate the input values of the feedforward controller algebraically with (9)–(13).

C. Program Structure of the Flatness-Based FFC

The above shown methods and algorithms lead to an efficient implementation of the controller. In a Siemens PLC, programing language S7-SCL allows a modular concept. Thus, there is one main program that calls the individual functions according to their moment of application. As shown in Fig. 7, a preselection determines if the controller faces a constant working point or a

Algorithm 2: Calculate Fifth-Order Polynomial at Point x.

$sum = 0$
$\alpha = (\eta + 1)$
for $i = 1$ to $(\eta + 1)$ by 1 **do**
  $\alpha = \alpha - 1$
  // Calculate power function
  $x_{\text{temp}} = x^{\alpha}$
  // Calculate index of coefficient
  $id = 4((i_m - 1) + m((i_s - 1) + s(i_\beta + i(\beta + 1))))$
  // Calculate element of polynomial
  $g = u_{\text{temp}} \cdot coeffSpl.DD[id]$
  $InstDB.DD[4(i-1)] = g$
end for
// Calculate sum over all elements
for $j = 0$ to η by 1 **do**
  $sum = InstDB.DD[4j] + sum$
end for
$polynom(x) = sum$

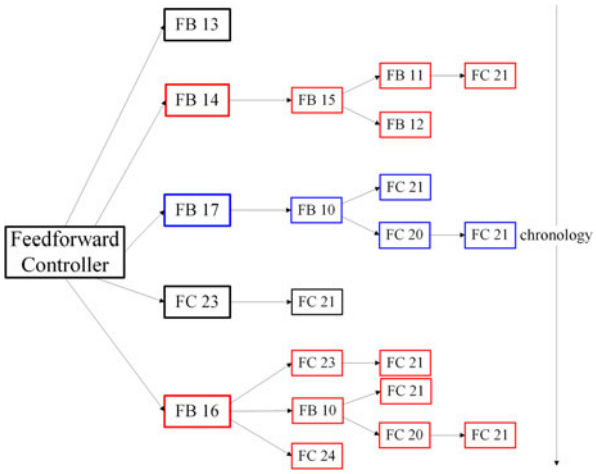

Fig. 7. Call up structure of PLC main program.

transition. While all black framed blocks are executed in every cycle, red framed blocks identify necessary functions in case of a transition and blue framed blocks ones for a constant working point. In detail, there are functions FC as well as function blocks FB, which differ in the fact that FBs include a memory data block. Table II assigns the executed functions to the called program parts.

VI. Simulation Results

To investigate the control performance, the results are compared with the ones presented in [15] where a MATLAB-/Simulink-based simulation has been employed. As the computation power of the control units is much smaller than the one from a standard PC, the implementation can lead to an undesirable control performance, which would make the new control approach unsuitable for practical use. During design and simulative verification of the new control algorithm, the real-time capability was not a matter of subject. As the approach consists of several complex and time-consuming mathematical operations, first of all these methods have to be verified in real-time operation. When comparing the results of the PLC calculations to the design calculations, the LU decomposition in order to solve (16) for the trajectory planning yields a relative error smaller than $10^{-3}\%$ within the vector of the order of 12. Observing the following evaluation of the fifth-order polynomials, the relative error still is smaller $10^{-1}\%$. It is important to notice that even the combination of all mathematical operations does not exceed the sampling time. Hence, we can prove these results to be reliable if we take a look to the numerical properties and stability. By division of the compressor map approximation, we can show that the condition number are smaller than ten for all feasible input sets, e.g., the error will always be of same magnitude. To show robustness against error propagation as well as cancellation errors, we verified the stability for each of the algorithms. For an example of the used methods, see to derived stability of the LU decomposition. The above shown very small relative errors can also be considered from a physical point of view. In the real application, the scaling of the IGV input in the interval $[0, 1]$ represents the range from 0 to 110. The resulting absolute error in determination of these input values will be lower than 0.001, which is significantly lower than the precision of the actuators. Also, the implementation of each mathematical operation for the trajectory calculation with an accuracy of 32bits has been verified. Analyzing the results shows that the difference between the designed approach, which is implemented in MATLAB, and the real-time realization is less than 4%. This leads to the conclusion that the proposed flatness-based FFC can be implemented with a negligible error on a PLC.

TABLE II
Functions and Function Blocks in PLC

Block	Name	Function
FB 10	GetIGV	Approximates the compressor map
FB 11	FillLS	Aligns the linear system
FB 12	SolveLS	Solves the linear system
FB 13	Preselection	Decides if working is constant or not
FB 14	CoeffTrans	Determines the trajectories for a transition
FB 15	CalcSpline	Calculates spline coefficients
FB 16	CtrlInTrans	Calculates the control inputs for a transition
FB 17	CtrlInConst	Calculates the control inputs for a constant working point
FC 20	Bias	Calculates bias for approximation
FC 21	Pow	Calculates power function
FC 22	GetMyOP	Reads current working point
FC 23	Poly5	Solves fifth-order polynomial
FC 24	Offset	Calculates offset in case of control difference at the beginning of a transition

To demonstrate the performance, a typical load rejection case is shown here. The given boundary constraints for the flat output (14) trajectory are

$$\begin{aligned} p_s(t_0) &= p_{s0},\ p_s^{(i)}(t_0) = p_{s0}^{(i)},\ p_s^{(i)}(t_T) = 0,\ i = 1, \\ p_t(t_0) &= p_{t0},\ p_t^{(i)}(t_0) = p_{t0}^{(i)},\ p_t^{(i)}(t_T) = 0,\ i = 1,2,3, \\ w_c(t_0) &= w_{c0},\ w_c^{(i)}(t_0) = w_{c0}^{(i)},\ w_c^{(i)}(t_T) = 0\ i = 1,2. \end{aligned} \quad (21)$$

As comparative values, the three control values named compressor mass flow, w_c, the suction pressure, p_s, and the GT inlet pressure, p_t, as well as the calculated FFC inputs, $\mathbf{u}$, are considered in an interval around the operating point change

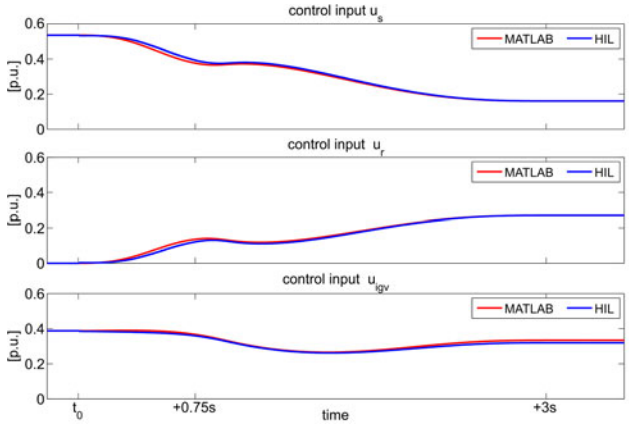

Fig. 8. Comparison of the FFC inputs.

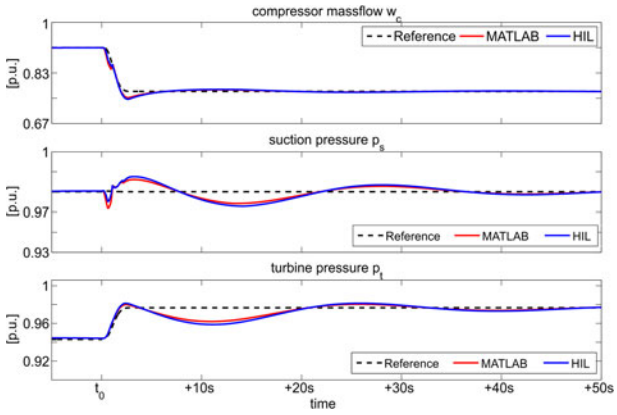

Fig. 9. Comparison of set values and real values.

$t \in [t_0, t_T]$. Fig. 8 compares the calculated FFC inputs. The red line indicates the simulation results of the control design [15] while the blue line shows the results of the HIL-simulation.

Fig. 9 shows the time behavior of the control values w_k, p_s, and p_t. The red line shows the simulation results of MATLAB and the blue one the HIL-simulation results. The particular dashed line specifies the reference signal of both simulations. Since all the relevant changes in the control input arise during the transition, these values are shown in Fig. 8 only in this interval, while the control values are shown until the new operating point is reached.

In order to prove the stability of the compressor during the transition, the compressor map is presented in Fig. 10. The red-dashed line indicates the reference transition, while the results of the MATLAB simulation are shown in red and the HIL-simulation values in blue.

Analyzing the signals in Figs. 8 to 10, the difference between the control design in MATLAB and the real-time realization on the HIL-platform can be seen to be minimal. This indicates that the new control approach can be implemented on a related control unit despite the low computation power and thus, leads to a high potential for a practical application of this advanced control technique. Note that for the real process t_0 will be given by the operator. Comparing these results (see Fig. 10) with the performance of the current decentralized PI controllers (see Fig. 3), it can be shown that the new control design is able to improve the control performance.

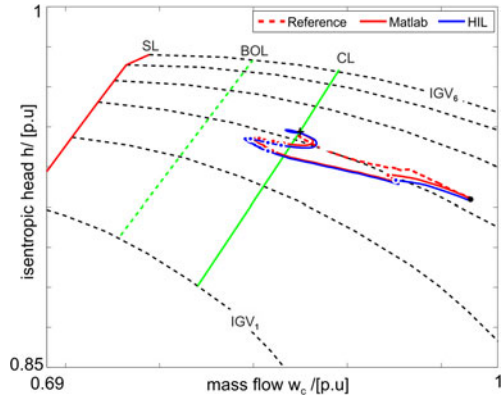

Fig. 10. Simulation of a load rejection with the new control approach.

VII. Conclusion

As shown in Fig. 3, the current PID controller of a fuel-gas compressor system has a bad performance during fast load rejections. By extending the control structure by the designed flatness-based FFC approach, fast load rejections can be handled with a much better performance. The objective of this paper was to show the real-time realization of a flatness-based FFC for a fuel-gas compressor system. The flatness-based FFC is characterized by its high standard of mathematical operations, which is rarely used in practice. As part of this work, numerical methods such as the solution of linear systems of equations or the evaluation of polynomials were successfully implemented. The low 32 bits arithmetic accuracy of the PLC was compensated by using an LU decomposition with pivoting, and a correction of operational sequences was introduced. The implemented FFC was successfully verified by the comparison with the results of [15]. This shows that the presented advanced control technique has the potential for practical use. Fast operating point transients of the compressor system can now be handled. The main benefit of the new control approach is that it does not replace the currently existing controllers. Since the feedforward controller simply extends the control loop, it can be implemented on a real plant without taking a risk.

References

[1] R. Vilanova and A. Visioli, *PID Control in the Third Millennium Lessons Learned and New Approaches*. London, U.K.: Springer, 2012.
[2] K. J. Åström and T. Hägglund, "The future of PID control," *Control Eng. Practice*, vol. 9, no. 11, pp. 1163–1175, Nov. 2001.
[3] G. Gu, A. Sparks, and S. S. Banda, "An overview of rotating stall and surge control for axial flow compressors," *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 6, pp. 639–647, Nov. 1999.
[4] F. Willems and B. De Jager, "Modeling and control of compressor flow instabilities," *IEEE Control Syst. Mag.*, vol. 19, no. 5, pp. 8–18, Oct. 1999.
[5] E. M. Greitzer, "Surge and rotating stall in axial flow compressors: Part 1 and 2l," *J. Eng. Gas Turbines Power*, vol. 98, no. 2, pp. 190–198, Apr. 1976.
[6] F. K. Moore and E. M. Greitzer, "A theory of post-stall transients in axial compression systems: Part 1-development of equations," *J. Eng. Gas Turbines Power*, vol. 108, no. 1, pp. 68–76, Jan. 1986.
[7] J. Simon, L. Valavani, A. Epstein, and E. Greitzer, "Evaluation of approaches to active compressor surge stabilization," *J. Turbomach.*, vol. 115, no. 1, pp. 57–67, Jan. 1993.
[8] O. Badmus, S. Chowdhury, and C. Nett, "Nonlinear control of surge in axial compression systems," *Automatica*, vol. 32, no. 1, pp. 59–70, Jan. 1996.

[9] D. Fontaine, S. Liao, J. Paduano, and P. V. Kokotovic, "Nonlinear control experiments on an axial flow compressor," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 5, pp. 683–693, Sep. 2004.
[10] M. Krstic, D. Fontaine, P. V. Kokotovic, and J. D. Paduano, "Useful nonlinearities and global stabilization of bifurcations in a model of jet engine surge and stall," *IEEE Trans. Autom. Control*, vol. 43, no. 12, pp. 1739–1745, Dec. 1998.
[11] J. T. Gravdahl and O. Egeland, "Centrifugal compressor surge and speed control," *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 5, pp. 567–579, Sep. 1999.
[12] J. T. Gravdahl, O. Egeland, and S. O. Vatland, "Drive torque actuation in active surge control of centrifugal compressors," *Automatica*, vol. 38, no. 11, pp. 1881–1893, Nov. 2002.
[13] F. Blanchini and P. Giannattasio, "Adaptive control of compressor surge instability," *Automatica*, vol. 38, no. 8, pp. 1373–1380, Aug. 2002.
[14] L. Giarré, D. Bauso, P. Falugi, and B. Bamieh, "LPV model identification for gain scheduling control: An application to rotating stall and surge control problem," *Control Eng. Practice*, vol. 14, no. 4, pp. 351–361, Apr. 2006.
[15] S. Osmic, M. Berner, A. Schwung, M. Jost, and M. Monnigmann, "Flatness-based feedforward control for fast operating point transitions of compressor systems," in *Proc. IEEE Conf. Control Appl.*, 2014, pp. 1753–1758.
[16] A. Schwung, P. Berner, S. Osmic, M. Jost, and M. Monnigmann, "Nonlinear decoupling control of compressor systems for fast load transients in combined cycle power plants," in *Proc. IEEE Conf. Control Appl.*, 2014, pp. 27–32.
[17] S. Osmic and A. Trachtler, "Flatness-based online controller reconfiguration," in *Proc. 34th Annu. Conf. IEEE Ind. Electron. Soc.*, 2008, pp. 204–209.
[18] 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.
[19] A. Gensior, H. Sira-Ramirez, J. Rudolph, and H. Guldner, "On some nonlinear current controllers for three-phase boost rectifiers," *IEEE Trans. Ind. Electron.*, vol. 56, no. 2, pp. 360–370, Aug. 2008.
[20] A. Houari, H. Renaudineau, J.-P. Martin, S. Pierfederici, and F. Meibody-Tabar, "Flatness-based control of three-phase inverter with output LC filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2890–2897, Oct. 2011.
[21] A. Shahin, M. Hinaje, J.-P. Martin, S. Pierfederici, S. Rael, and B. Davat, "High voltage ratio dc–dc converter for fuel-cell applications," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 3944–3955, Nov. 2010.
[22] J. Linares-Flores, A. HernandezMendez, C. Garcia-Rodriguez, and H. Sira-Ramirez, "Robust nonlinear adaptive control of a "boost" converter via algebraic parameter identification," *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4105–4114, Feb. 2014.
[23] G. Rigatos, P. Siano, and N. Zervos, "Sensorless control of distributed power generators with the derivative-free nonlinear kalman filter," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6369–6382, Jun. 2014.
[24] H. Aschemann and D. Schindele, "Sliding-mode control of a high-speed linear axis driven by pneumatic muscle actuators," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3855–3864, Oct. 2008.
[25] S. Yin and X. Zhu, "Intelligent particle filter and its application to fault detection of nonlinear system," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3852–3861, May 2015.
[26] S. Yin and Z. Huang, "Performance monitoring for vehicle suspension system via fuzzy positivistic c-means clustering based on accelerometer measurements," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2613–2620, Jul. 2015.
[27] S. Dominic and U. Maier, "Dynamic modeling and simulation of compressor trains for an air separation unit," in *Proc. 19th IFAC World Congr.*, Cape Town, South Africa, 2014, pp. 432–437.
[28] J. Kocian and J. Koziorek, "An outline of advanced process control and self tuning techniques on PLC background," in *Proc. 16th IEEE Conf. Emerging Technol. Factory Autom.*, Bilbao, Spain, 2010, pp. 1–8.
[29] S. Skoczowski, S. Domek, K. Pietrusewicz, and B. Broel-Plater, "A method for improving the robustness of PID control," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1669–1676, Dec. 2005.
[30] C. Rossi and A. Tonielli, "Robust current controller for three-phase inverter using finite-state automaton," *IEEE Trans. Ind. Electron.*, vol. 42, no. 2, pp. 169–178, Aug. 1995.
[31] J. Kocian and J. Koziorek, "Implementation of fuzzy logic control based on plc," in *Proc. 16th IEEE Conf. Emerging Technol. Factory Autom.*, Toulouse, France, 2011, pp. 1–8.
[32] S. Giannoutsos and S. Manias, "A data-driven process controller for energy-efficient variable-speed pump operation in the central cooling water system of marine vessels," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 587–598, Dec. 2015.
[33] W. Goff, "Automation of reciprocating gas engine compressor packages using programmable logic controllers," *IEEE Trans. Ind. Appl.*, vol. 26, no. 5, pp. 909–913, Sep. 1990.
[34] M. Zeitz, "Vorsteuerungsentwurf im Frequenzbereich: Offline oder online," *at-Automatisierungstechnik*, vol. 60, no. 7, pp. 375–383, Jul. 2012.
[35] M. Zeitz, "Flachheitsbasierter Entwurf linearer zeitvarianter SISO-Systeme," *at-Automatisierungstechnik*, vol. 58, no. 7, pp. 351–360, Jul. 2010.
[36] R. Rothfuß, J. Rudolph, and M. Zeitz, "Flachheit: Ein neuer Zugang zur Steuerung und Regelung nichtlinearer Systeme," *at-Automatisierungstechnik*, vol. 45, no. 11, pp. 517–525, Nov. 1997.
[37] G. Allaire, *Numerical Linear Algebra* (Texts in Applied Mathematics), vol. 55. New York, NY, USA: Springer, 2008.

Shane Dominic received the Ph.D. degree in electrical engineering from the University of Duisburg-Essen, Duisburg, Germany, in 2016.

He is currently an R&D Engineer for Toyota Motorsports GmbH, Cologne, Germany. His research interests include model-based control, economic process optimization, neural network-based nonlinear system identification, and their application to industrial systems.

Yannik Löhr received the B.Sc. degree in electrical engineering and the M.Sc. degree in automation from the University of Duisburg-Essen, Duisburg, Germany, in 2013 and 2015, respectively. He is currently working toward the Ph.D. degree at the Ruhr-University Bochum, Bochum, Germany.

His research interests include model-based control and predictive control of thermal, hydraulic, and electrical systems.

Andreas Schwung received the Ph.D. degree in electrical engineering from the Technische Universität Darmstadt, Darmstadt, Germany, in 2011.

From 2011 to 2015, he was an R&D Engineer with MAN Diesel & Turbo SE, Oberhausen, Germany. Since 2015, he has been a Professor of automation technology at the South Westfalia University of Applied Sciences, Soest, Germany. His research interests include model-based control, networked automation systems, and intelligent data analytics with applications in manufacturing, process industry, and electromobility.

Steven X. Ding received the Ph.D. degree in electrical engineering from the Gerhard-Mercator University of Duisburg, Duisburg, Germany, in 1992.

From 1992 to 1994, he was an R&D Engineer with Rheinmetall GmbH. From 1995 to 2001, he was a Professor of control engineering with the University of Applied Science Lausitz, Senftenberg, Germany, and served as a Vice President of this university during 1998–2000. Since 2001, he has been a Professor of control engineering and the Head of the Institute for Automatic Control and Complex Systems, University of Duisburg-Essen, Essen, Germany. His research interests are model-based and data-driven fault diagnosis, fault-tolerant systems, and their application in industry with a focus on automotive systems and chemical processes.