Field-Programmable System-on-Chip-Based Control System for Real-Time Distortion Correction in Optical Imaging

Hannes Radner, Johannes Stange, Lars Büttner, and Jürgen Czarske, *Senior Member, IEEE*

***Abstract*—The digital transition requires real-time control of complex systems with short loop time and low latency in various applications. Field-programmable gate arrays (FPGAs) are, in principle, capable of complying with this task but demand, on the other hand, a high programming effort. In this article, we propose a field-programmable system on chip (FPSoC) as a hybrid solution of an FPGA and a central processing unit (CPU) on a single monolithic die to combine the strengths of both architectures. An FPSoC-based adaptive optical wavefront correction system is presented as a case study to correct camera images in real time that are distorted by time-varying aberrations. While a short total loop time is achieved by interfacing the camera and a deformable mirror on a low level directly with the FPGA, all computationally nonintensive tasks are implemented on the CPU to keep the flexibility, reusability, and development expense low. The system corrects the optical distortion of water surface waves with up to 3600 control cycles per second and spatially attenuates the distortion up to Zernike polynomial 14 with up to 150 Hz. The FPSoC system enables fast spatiotemporal aberration correction in technical processes and offers a perspective for measuring complex flows through fluctuating interfaces.**

***Index Terms*—Adaptive optics, field-programmable system on chip (FPSoC), multiple-input multiple-output systems, wavefront correction.**

I. Introduction

Adaptive optical wavefront correction systems play a central role in an increasing number of applications, such as in earth-bound telescopes, ophthalmology, microscopy, or long-range optical free-space communication. However, no universal solution exists to achieve a real-time control of such complex optomechatronic systems with a short loop time. Accomplishing a loop time below 1 ms is challenging because of the long complex loop sequence.

The control system needs to capture an image with a wavefront sensor, process the data to retrieve the wave front, decompose the wavefront into an orthonormal base, calculate the optimum amplitude settings for compensation, and send them to the adaptive optical correction element. The computational latency can be tackled by high-performance systems with multiple central processing units (CPU) accelerated by graphics processing units (GPU) [1], but a large amount of the total loop time is caused by the image acquisition, the frame grabber, and the interface to control the adaptive optical element. A straightforward solution would be to use a field-programmable gate array (FPGA) with a direct low-level interface to the external components. This approach was applied for adaptive optics [2], and it is also widely deployed in industrial applications [3], [4] or to realize model-predictive controllers [5]–[9] or real-time controlling tasks [10]–[13], cost-sensitive applications [14], or whether a lot of preprocessing of, e.g., camera or sensor data is needed [15]–[18].

However, FPGA-based systems have some drawbacks over CPU- and GPU-based systems [19]. FPGAs are not programmed directly. Instead, a hardware descriptive language, e.g., Verilog, is used to describe the behavior of the hardware, or special high-level synthesis tools are used to convert, for example, C-code into a hardware-descriptive language. This behavioral description is then synthesized into a register-transfer-level (RTL) schematic. In the next stage, the design toolchain places and routes the RTL schematic into a specific FPGA chip. This process can take up to several hours even for very small changes in the design. This is not feasible for the development, test, debugging, comparison, or parameterization and highly increases the development expense and costs. High-level tasks or interfaces are also difficult to implement. One example is the implementation of an ethernet interface in order to control the FPGA-based controller via the transmission control protocol/Internet protocol over the network. This is essential, e.g., for industry 4.0, where all controlling devices of a factory are connected with a standardized interface [20].

To overcome these drawbacks, we propose to deploy field-programmable system-on-chips (FPSoCs) as a combination of a fast native CPU and an FPGA on one chip with a shared memory as a hybrid solution. These devices with dedicated CPUs are much more powerful than soft cores synthesized into

Manuscript received September 27, 2019; revised December 23, 2019 and February 3, 2020; accepted February 25, 2020. Date of publication March 16, 2020; date of current version December 8, 2020. This work was supported by the Deutsche Forschungsgemeinschaft within a Reinhart Koselleck projekt under Grant CZ55-30. *(Corresponding author: Hannes Radner.)*

The authors are with the Faculty of Electrical and Computer Engineering, Laboratory of Measurement and Sensor System Technique, Technische Universität Dresden, 01062 Dresden, Germany (e-mail: hannes.radner@tu-dresden.de; johannes.stange@mailbox.tu-dresden.de; lars.buettner@tu-dresden.de; juergen.czarske@tu-dresden.de).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

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

Digital Object Identifier 10.1109/TIE.2020.2979557

the FPGA and save FPGA resources. In this article, we show in a case study at an optical wavefront correction system that it is possible with an FPSoC to achieve a short loop time by interfacing the hardware on a low level directly with the FPGA. All computationally nonintensive tasks are implemented on the CPU to keep the flexibility, reusability, and development expense low.

The system presented in our case study is used to compensate the optical distortion induced by a fluctuating water–air interface. This is a challenging task, since the surface fluctuates with up to several hundred Hertz. To meet this requirement, we combined a fast deformable membrane mirror with a wavefront sensor and a flexible signal processing system based on an FPSoC (Zynq-7100, Xilinx).

The rest of this article is organized as follows. In Section II, the optical setup is presented and the Fresnel Guide Star (FGS) is introduced as a novel feedback value. In Section III, the implementation of the hybrid control loop on the FPSoC chip is explained in detail. In Section IV, the performance of the control system is analyzed by means of two experiments. Finally, Section V concludes this article.

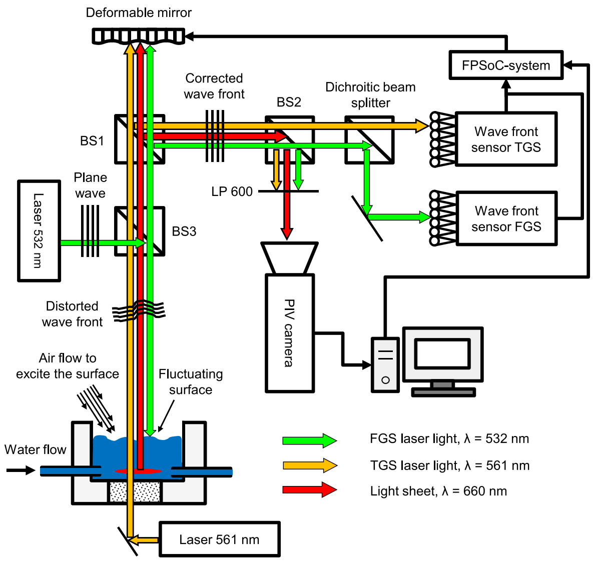


Fig. 1. Schematic of the optical setup of the presented wavefront correction system with two different guide star approaches. LP—long-pass filter; PBS—polarizing beamsplitter; BS—50:50 beamsplitter; TGS—transmission guide star; FGS—Fresnel guide star.

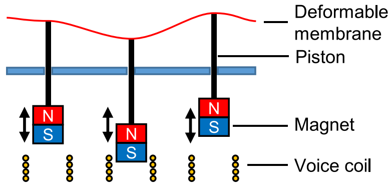
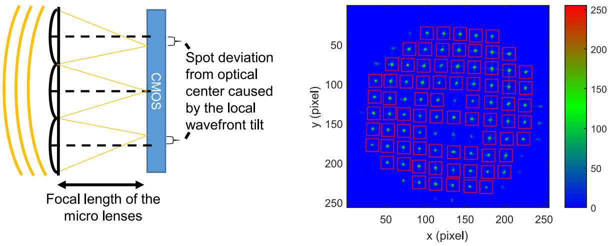


Fig. 2. Schematic working principle of a deformable membrane mirror.


CMOS
Spot deviation from optical center caused by the local wavefront tilt
Focal length of the micro lenses
50
100
150
200
250
y (pixel)
x (pixel)
Intensity (AU)
0

(a) (b)

Fig. 3. (a) Schematic working principle of a Hartmann–Shack wavefront sensor. (b) Hartmannogramm captured by the CMOS chip. The red boxes mark the area of the center-of-mass evaluation for the spot tracking.

II. Control System for Adaptive Optical Wavefront Correction

A. Optical Setup

As the test case, we consider a camera-based flow measurement to be conducted through a fluctuating air–water phase boundary. The velocity information is derived from particles carried with the flow, which are tracked by a camera (particle image velocimetry—PIV). The wavefront correction system will be employed to compensate for the image distortion caused by light refraction at the interface. The optical setup of the measurement system, including adaptive optical correction, is shown in Fig. 1. The measurement object is a water-filled basin, where a flow can be generated behind a nozzle. To excite capillary waves on the open water surface, an airflow is used. The airflow does neither influence the flow behind the nozzle nor does the water flow contribute to the surface fluctuation. The basin has two windows: one at the bottom for a transmission guide star (TGS) (561 nm) and a side window for a laser light sheet (660 nm) for PIV measurements (see Section IV-B). The light from the TGS and the PIV light propagate through the fluctuating surface and are affected identically by the distortion. To correct for optical distortions with single optical access from the top, a third laser (532 nm) for the FGS as an alternative to the TGS is implemented. In contrast to the TGS, the optical distortion is measured by evaluating approximately 7% of light, which is reflected back from the surface. This Fresnel reflex contains all information about the distortion (see Section II-B). The reflected light propagates backward through the beamsplitter BS3 toward the deformable mirror (DM69, Alpao, France). The mirror consists of a deformable metallic membrane, which can be actuated by 69 individual pistons below the membrane (see Fig. 2). A single actuator at the center of the mirror reaches a phase lag of −45° at 1261 Hz. With the 69 pistons, a superposition of the first 14 Zernike polynomials [21] can be displayed. Zernike polynomials define an orthonormal base, which are commonly used to describe optical wavefront distortions, e.g., tilt, defocus, or astigmatism. A linear combination of the polynomials is used to display the inverse distortion of the water surface on the mirror and thereby correct the wavefront. Beamsplitter BS2 directs a part of the light from the light sheet onto a camera (PIV camera) to record the particle distribution in the flow. The FGS and the TGS are blocked by an optical long-pass filter. The light from both guide stars is separated by a dichroic beamsplitter and directed onto a fast in-house developed Hartmann–Shack wavefront sensor. The basic working principle of the sensor is

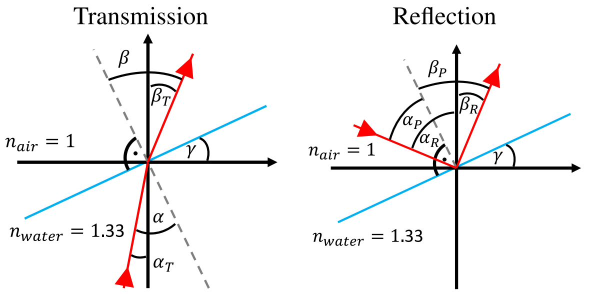


Fig. 4. Comparison of the beam path of a transmitted beam used for the TGS and a reflected beam used for the FGS. Blue line—tilted water surface; dashed line—surface normal; red line—beam path.

shown in Fig. 3(a). It consists of a microlens array and a complementary metal–oxide–semiconductor (CMOS) camera placed in its focal plane. The wavefront is sampled by the microlenses. The resulting Hartmannogramm is shown in Fig. 3(b). A tilt of the local wavefront leads to a displacement of the spot on the CMOS sensor from the optical center of the lens. The position of the spots in the Hartmannogramm is used to decompose the wavefront in low-order Zernike polynomials, which is further discussed in Section III-A. The CMOS chip (LUPA3000 A, On Semiconductor) is able to capture about 10 000 frames/s with a resolution of 256 px × 256 px. The chip directly sends the image data into the FPGA part of the FPSoC for the lowest possible transmission latency. The FPSoC system evaluates the image data from the sensor and controls the deformable mirror accordingly.

B. Fresnel Guide Star

In contrast to a guide star observed in transmission, the novel FGS technique [22]–[24] additionally implemented in the presented setup (see Fig. 1) uses the reflected light of the surface to measure the optical distortion induced by the interface. This enables a measurement of the distortion with single optical access. Using the reflex as the feedback value poses an extra challenge for the control loop, since in transmission, the controlled value equals the feedback value, but in reflection, it does not. The reasons for this are the different laws for refraction and reflection. If the Fresnel Reflex of the surface is used as the feedback signal, the system will need to recover the control value from it. The theory of the beam refraction at the surface is illustrated in Fig. 4. The angle β_T of the transmitted beam is determined by Snell's law

$$\frac{\sin\beta}{\sin\alpha} = \frac{n_{\text{water}}}{n_{\text{air}}}. \tag{1}$$

With respect to the local tilt angle γ of the surface, β_T can be determined as

$$\beta_T = \arcsin\left(\sin\left(\alpha+\gamma\right) * \frac{n_{\text{water}}}{n_{\text{air}}}\right) - \gamma. \tag{2}$$

For the Fresnel Reflex, the incident angle α_P equals the reflected angle β_P with respect to the surface normal. With this relationship, the angle β_R can be calculated as

$$\beta_R = \alpha_R - 2\gamma. \tag{3}$$

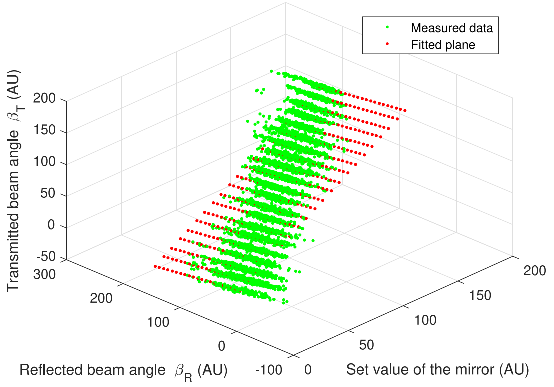


Fig. 5. Method of calibration for the control value recovery. A plane is fitted to the measured data, which are used by the loop to recover the control value from the FGS feedback value.

This relationship is not just valid for single beams, but also for two-dimensional Zernike polynomials. A limitation of this recovery strategy is that height changes of the surface are neglected, but the results of Section IV proof that the neglection is tolerable for small air-induced surface waves. For the setup presented in Fig. 1, the incident beam angles α_T and α_R are kept constant. Since the deformable mirror modulates the wavefront before it is measured by the Hartmann–Shack sensor, its set value also has an influence to the measured feedback value. Consequently, the control value β_T is a linear combination of the mirror set value and the angle of the Fresnel Reflex β_R and can be written as

$$\beta_T = A * \text{setValue} + B * \beta_R + C. \tag{4}$$

It is necessary to include the mirror in the loop for the calibration process of the Zernike decomposition, which is discussed in Section III-A. Regarding the experimental setup, it is not possible to numerically calculate the values for A, B, and C because the exact position and orientation of all components in the setup cannot be determined. Instead, a calibration procedure is executed, in which the mirror is set to different set values, and β_R and β_T are measured simultaneously for random surface angles γ. Fig. 5 shows a result of the calibration as an example. The parameters A, B, and C were determined by a least-squares fit of a plane to the data. This linear fit is only valid for small tilt angles.

C. Hardware

In order to achieve a high attenuation bandwidth for the distortion compensation, a specialized processing system is developed. The design goal for future measurements is the compensation of optical distortions of a fluctuating air–water interface with only one optical access, e.g., to enable measurements inside droplets or bubbles. Droplets with 15–20 μl can fluctuate with up to 150 Hz [25], [26]. Together with the demand of correcting Zernike polynomials up the 14th order, as determined by the deformable mirror, this leads to the following requirements.

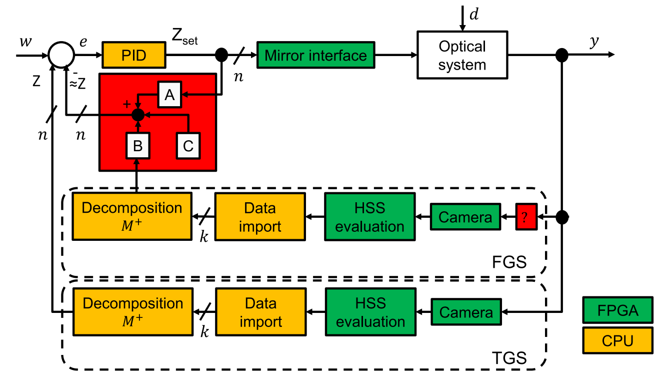


Fig. 6. Flowchart of the implemented control loop. The red boxes mark the special difference between the FGS and the TGS. The feedback signal is only related to the control value, and the actual control value has to be recovered. Signal of TGS—transmission guide star; signal of FGS—Fresnel guide star; k—number of spots on the Hartmannogramm; n—number of Zernike polynomials.

1) The total loop time should be less than 660 μs to achieve a factor of ten as reserve.
2) The Hartmann–Shack wavefront sensor has to determine the first 14 Zernike polynomials of the FGS wavefront from a 256 × 256 px image with 30–100 spots.
3) The control system has to recover the control value from the FGS wavefront and set the deformable mirror to compensate the distortion.

The FPSoC-based control system was developed in-house to achieve the design goal with a maximum flexibility. It consists of four stacked printed circuit boards (PCBs). The first one is for the power management, the second serves as an interconnect backplane, and the third one implements the hardware-specific interface to the mirror and to the cameras. The fourth PCB is an FPSoC-Module TE0782-02-100-2I from Trenz Electronic Germany, which features a Zynq-7100 FPSoC from Xilinx. This ready-to-use module implements all basic circuits and interface chips for the FPSoC and highly decreases the development expense.

III. Implementation of the Control Loop on the FPSoC

A. Control Loop

Fig. 6 shows the high-level flowchart of the control loop. To make an efficient use of the FPSoC's architecture, some parts are implemented in the FPGA, while others are implemented in a C++ application running in the user space of a Linux operating system. As a feedback value, either the TGS or the FGS can be used. The evaluation of the Hartmannogramm is comparable for the TGS and the FGS. First, the position of all spots on the Hartmannogramm [see Fig. 3(b)] is determined. The k spot positions are decomposed into a superposition of n Zernike polynomials by means of a matrix multiplication. It performs a least-squares fit of a linear combination of Zernike polynomials to the measured spot positions. The matrix is obtained in a calibration process by displaying one Zernike Z at a time on the mirror and saving the deviation of the spots S column by column in the M matrix [see (5)]. The Moore–Penrose inverse M^+ is then calculated by (6):

$$\begin{pmatrix} M_{Z_1S_1} & M_{Z_2S_1} & M_{Z_3S_1} & \cdots \\ M_{Z_1S_2} & M_{Z_2S_2} & M_{Z_3S_2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \cdot \begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ \vdots \end{pmatrix} \tag{5}$$

$$M^+ = (M^*M)^{-1}M^*. \tag{6}$$

For the FGS, the control value Z is retrieved by solving (4), as discussed in Section II-B. The measured Zernike combination is compared with a plane reference wavefront w, and the deviation e is fed into n proportional–integral–differential (PID) controllers. The Zernike outputs of the PID controllers are superposed, and the needed control values are sent to the 69 actuators of the mirror. The mirror compensates the optical distortion d and approaches a corrected wavefront y.

B. Implementation on the FPSoC

The detailed implementation of the FPGA part is shown in Fig. 7. The design is organized in four different main modules and consists of seven separated clocking domains.

1) Processing System: The processing system (PS) consists of two ARM A9 cores, a central interconnect bus, attached peripherals, a connected DDR 3 memory, and advanced extensible interface (AXI) bus connections to interface the programmable logic. The Zynq-7100 features two types of AXI buses: the high-performance HP-AXI, which is designed for a high data throughput, and the general-purpose GP-AXI, which is optimized for low-latency control tasks. The HP-AXI interface is only used to send video data to the high-definition multimedia interface (HDMI). The GP-AXI interface is expanded by an AXI interconnect module from Xilinx to interface the various AXI devices in the design.

This bus creates a memory space for all the connected slaves. The address space of the memory starts with CPU and special function registers to control the hardware of the processing system. This is followed by the address space of the double data rate (DDR) memory with the Linux Kernel and the user space with user applications. The residual memory space can be used to address the logic in the FPGA portion. There are several strategies for the C++ application to communicate with the logic in the FPGA, e.g., writing a kernel module, using a device driver or memory mapping. For control tasks, memory mapping is the best strategy to ensure a low latency. Therefore, a small dual-port memory is synthesized into the FPGA. One port of the block random access memory (BRAM) is connected to the PS with the AXI Bus and an AXI BRAM Controller from Xilinx. The other port is connected to the logic in the FPGA. When the application is started, the base address of the BRAM is stored to a pointer in the C++ application. In this way, the BRAM can be read and written to as it would be a normal variable or array with a comparable latency. To trigger or read signals from the

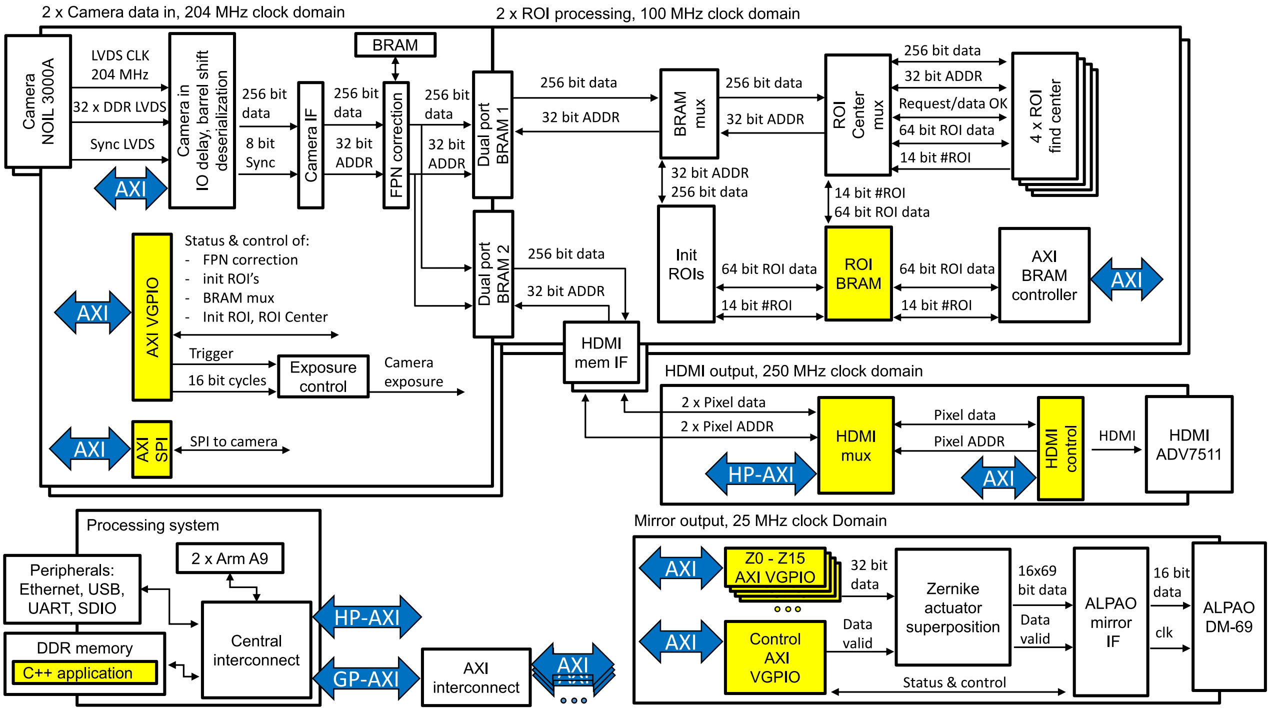


Fig. 7. Structure of the implemented design on the system-on-chip device. The yellow modules share one logical address space.

FPGA, the same strategy can be used with the difference that the storage cells are the direct output or input of the module. Xilinx provides a ready-to-use module named AXI GPIO as a virtual general-purpose input–output (VGPIO) for this purpose.

2) Data Processing: To evaluate the Hartmannogramm of the FGS and the TGS, the Camera data in and the region-of-interest (ROI) processing module are instantiated twice. The camera is connected by 32 double-data-rate low-voltage differential signaling (DDR LVDS) data channels, one DDR LVDS sync channel, and one 204-MHz LVDS clock input to the FPGA fabric. Since the camera defines the data rate, the whole camera data in clock domain is directly driven by the camera clock. Two dual-port BRAMs are used to decouple the cross clock domain data path to the ROI processing module. The camera transmits the pixel data serially in groups of 32 pixels (one kernel) with a packet size of 8 bits per pixel and signalizes the frame start, row start, and column on the sync channel. The Zynq-7100 offers dedicated LVDS IO hardware, which is used to deserialize the data stream to a 256-bit parallel data stream and an 8-bit parallel sync stream. To achieve a proper deserialization, the sampling point is centered to the middle of the eye pattern of a 0xAA training pattern by adjusting the channel delay. In a next step, a barrel shifter is used with a 0x0F training pattern to align the deserializer with the 8-bit pixel packet. The next camera IF module analyzes the sync channel for the frame start and row start signal to store each kernel with increasing address to the dual-port BRAMs. Before the kernel is saved, the fixed pattern noise (FPN) is corrected by subtracting the pixel offset. When the last kernel is received, the FPN module signals the AXI VGPIO that the image has been completely received.

The number of clock cycles to transfer an image from the CMOS to the dual-port BRAM is determined by the CMOS chip plus three control cycles needed by the FPN correction and the camera IF module. For an image size of 256 × 256 px, 17 960 clock cycles are needed in total.

The ROI processing module evaluates the Hartmannogramm. For the initial determination of the spot positions in the Hartmannogramm, the Init ROI module is used. The module analyzes each kernel of the image and saves a possible spot position if the gray level of a pixel exceeds 50%, and no other position within a defined distance was saved as a potential spot position. The positions are saved to the ROI BRAM, where the 14-bit address corresponds to the ROI number and the x and y position is saved as a 32-bit fixed point number with 16-bit integer part and 16-bit fractional part for subpixel resolution. When the control loop is in operation, the ROI center multiplexer (mux) module handles the parallel access of the four ROI find center modules to the ROI BRAM and the dual-port BRAM 1 by a request data and wait-in-line scheme. The saved spot positions in the ROI BRAM are divided into four equally sized groups. Each ROI find center module updates the last known position of an ROI saved in the ROI BRAM by a center-of-mass algorithm calculated in a small 15 × 15 px region at the last known position. This spot tracking enables a significant higher detection range for large distorted wavefronts measured by the Hartmann–Shack wavefront sensor. When the ROI find center modules finished the evaluation of the ROIs, they set a done flag, which is combined by an AND gate and signaled to an AXI VGPIO pin. The ROI BRAM is connected to an AXI BRAM controller to the processing system and can directly be accessed via memory mapping in the C++ application.

TABLE I
NEEDED CLOCK CYCLES PER MODULE OF THE DESIGN SHOWN IN FIG. 7

main Module	cycles	clock domain (MHz)	time (µs)
Camera data in	17960	204	88
ROI processing (85 spots)	12200	100	122
Mirror output	2300	25	92

TABLE II
RESOURCES USED BY THE DESIGN SHOWN IN FIG. 7

	available in Zynq®-7100	used	used (%)
LUT	277400	165946	59.8
LUTRAM	108200	3338	3
Flip flops	554800	182699	33
BRAM	755	743	98.5
DSP slices	2020	855	42.3

TABLE III
TIMING BUDGET FOR THE FLOWCHART IN FIG. 6 IN COMPARISON TO OTHER SYSTEMS REPORTED IN THE LITERATURE

	TGS system	FGS system	System [28]	System [1] ACE fast /ACE	System [2]
Platform	FPSoC	FPSoC	CPU	CPU & GPU /CPU	FPGA
number of spots	85	33	121	1600 /400	256
Image acquisition	97 µs	97 µs	unknown	sim /670 µs	4000 µs
HSS evaluation	122 µs	47 µs	40 µs	130 µs /1300 µs	4.9 µs
Data import	43 µs	17 µs			1.0 µs
Decomposition (M^+)	36 µs	13 µs			
Control value recovery	-	<1 µs			2.9 µs
Controller	5 µs	5 µs			
Mirror interface	92 µs	92 µs	unknown	sim /unknown	unknown
Σ (computational latency)	206 µs	83 µs	40 µs	130 µs /1300 µs	8.8 µs
Σ (time)	387 µs	271 µs	2000 µs	833 µs /1111 µs	4000 µs
Σ (frequency)	2.58 kHz	3.6 kHz	0.5 kHz	1.2 kHz /0.9 kHz	0.25 kHz

TGS: transmission guide star; FGS: Fresnel guide star; sim: not included in the loop simulation; ACE: ALPAO Core Engine.

The processing of one ROI takes 90 cycles in the worst case if an ROI splits over two horizontal adjacent kernels. Since the ROIs are evaluated in parallel, the multiplexed memory access is the bottleneck of this design. The ROI find center module usually has to wait in line a few cycles for the requested data and needs in average 143 cycles per ROI. On average, for the processing of a 256×256 px Hartmannogramm with 85 ROIs, 12 200 cycles are needed.

The LUPA3000A CMOS chip can be controlled by the AXI Quad SPI module from Xilinx to set, e.g., the DDR LVDS training pattern and analog-to-digital converter settings.

3) HDMI Output: The purpose of the module is to output the Linux screen of the processing system or the image captured by the cameras. The monitor is connected to an HDMI physical interface (phy) ADV7511 from Analog Devices, which is connected to the HDMI control module. The module was originally designed to only output the Linux screen. For this reason, an HDMI mux module was implemented. The HDMI output runs at 60 Hz with a resolution of 1080 p and is not synchronized to the cameras.

4) Mirror Output: The fourth main module is the mirror output module. It controls the deformable mirror from ALPAO. The module consists of 17 AXI VGPIO's. Sixteen of them output the desired Zernike coefficient between -1 and $+1$ of Z_0–Z_{15} in a 32-bit fixed point number with the 16-bit integer part and the 16-bit fractional part. When the desired Zernike coefficients are set, a data valid signal is sent to the module Zernike actuator superposition. This module calculates the superposition of the Zernike polynomials and the needed 16-bit integer values for the 69 actuators of the mirror. The result is handed over to the ALPAO mirror IF module by a 69×16 bit bus and a data valid signal. The ALPAO mirror IF module checks the data to protect the mirror from destructive set values and builds the packet stream. The data are transmitted via a 16-bit parallel bus to the mirror control box.

The module needs 18 clock cycles to calculate the superposition of the Zernike polynomials, 454 clock cycles to check and build the packet stream, and 1828 cycles to transmit the data at maximum possible bus speed. In total, 2300 cycles are needed.

Table I summarizes the resources used by the design and Table II the needed clock cycles per module. The implemented design was tested and verified with synthetic test data generated with a second adaptive mirror not shown in the optical setup.

C. C++ Implementation

The control loop is implemented in a C++ application, which runs in the user space of an embedded Linaro Linux operating system. The authors preferred to use Linux instead of a bare-metal or FreeRTOS environment, because there is a much larger support base, and at the design stage, it was found to be much easier to implement an HDMI in Linux, which is very useful for debugging purposes. The real-time capability is ensured by a careful software implementation and use of the CPU clock. The jitter of the loop time is smaller than 5 μs, which is negligible compared to the total loop times (see Table III).

The program sequence is shown in Fig. 8. The first step (2) of the C++ application is to initialize the modules in the FPGA part and the camera. In the next step, one image is captured, and the Init ROIs module is triggered to find the initial ROIs. The software waits for the module to signal the end of the Hartmannogramm evaluation with a high level at one of the VGPIOs. After that, the BRAM mux module is switched to give the ROI find center module access to the Hartmannogramm,

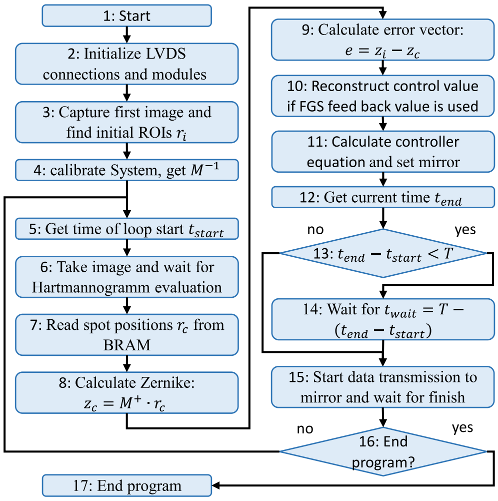

1: Start
2: Initialize LVDS connections and modules
3: Capture first image and find initial ROIs r_i
4: calibrate System, get M^{-1}
5: Get time of loop start t_{start}
6: Take image and wait for Hartmannogramm evaluation
7: Read spot positions r_c from BRAM
8: Calculate Zernike: $z_c = M^+ \cdot r_c$
9: Calculate error vector: $e = z_i - z_c$
10: Reconstruct control value if FGS feed back value is used
11: Calculate controller equation and set mirror
12: Get current time t_{end}
no
yes
13: $t_{end} - t_{start} < T$
14: Wait for $t_{wait} = T - (t_{end} - t_{start})$
15: Start data transmission to mirror and wait for finish
no
yes
16: End program?
17: End program

Fig. 8. Flowchart of the implemented C++ application.

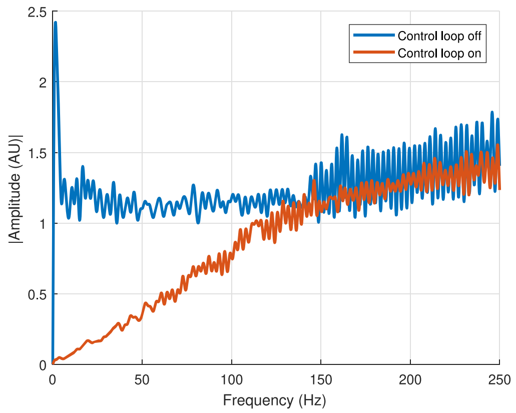


Fig. 9. Amplitude spectrum of the measured Zernike wavefront Z_1^1 (tilt) deviation e with the TGS control loop switched ON and OFF for a synthetic deflection of the TGS laser. The loop is able to attenuate distortions of up to 150 Hz.

which will analyze the images in all following steps. In step 4, the decomposition matrix M^+ is determined, as discussed in Section III-A. Step 5 is the first step of the actual control loop, where the start time of the CPU clock is saved. This will be necessary in step 14 to guarantee a well-defined loop cycle time with real-time properties. In step 6, an image is taken, and the Hartmannogramm is evaluated. The program waits for the ROI find center modules to signal the end of the Hartmannogramm evaluation with a high level at one of the VGPIOs. The next steps of the control loop require more complex math and are not computational demanding; thus, the position of all spots is now imported in step 7 to a vector of the Eigen C++ library [27] to the user space from the ROI BRAM. In step 8, the Zernike decomposition is done, as described in Section III-A. The error vector is calculated (step 9), and in case the FGS is used as a feedback source, the control value is recovered (step 10). The controller calculates the next set values in step 11 and applies them to the Zernike actuator superposition module. In the next three steps, the elapsed time is measured, and the program waits until a defined loop period time is reached to guarantee a well-timed loop cycle. In the last step 15 of the loop cycle, the ALPAO IF module is signalized to start the data transmission and waits for it to finish. A short minimum loop time of 271 μs can be reached for the FGS, which is important to attenuate fast distortions of up to 1845 Hz in theory.

Table III summarizes the timing budget for the main steps of the loop and compares it with the timing budget of other systems reported in the literature. Care must be taken when comparing the systems, because they use different strategies and algorithms, do not necessarily use a spot tracking, and may start to sample the next image while evaluating the last one. In general, it can be seen that the computational latency is not the crucial part of the timing budget and thus not the limitation for the maximum attenuation bandwidth. Most time is spent on capturing the image and interfacing the camera and mirror. Here, the FPSoC system has a major advantage, since it directly interfaces the camera chips and the mirror with the FPGA fabric, and it still offers the same flexibility and low programming effort of a CPU solution for the implementation of the actual control algorithms. Only for very large numbers of spots, it might be feasible to offload some C++ implemented functions to the FPGA on the price of flexibility and programming effort.

IV. Results

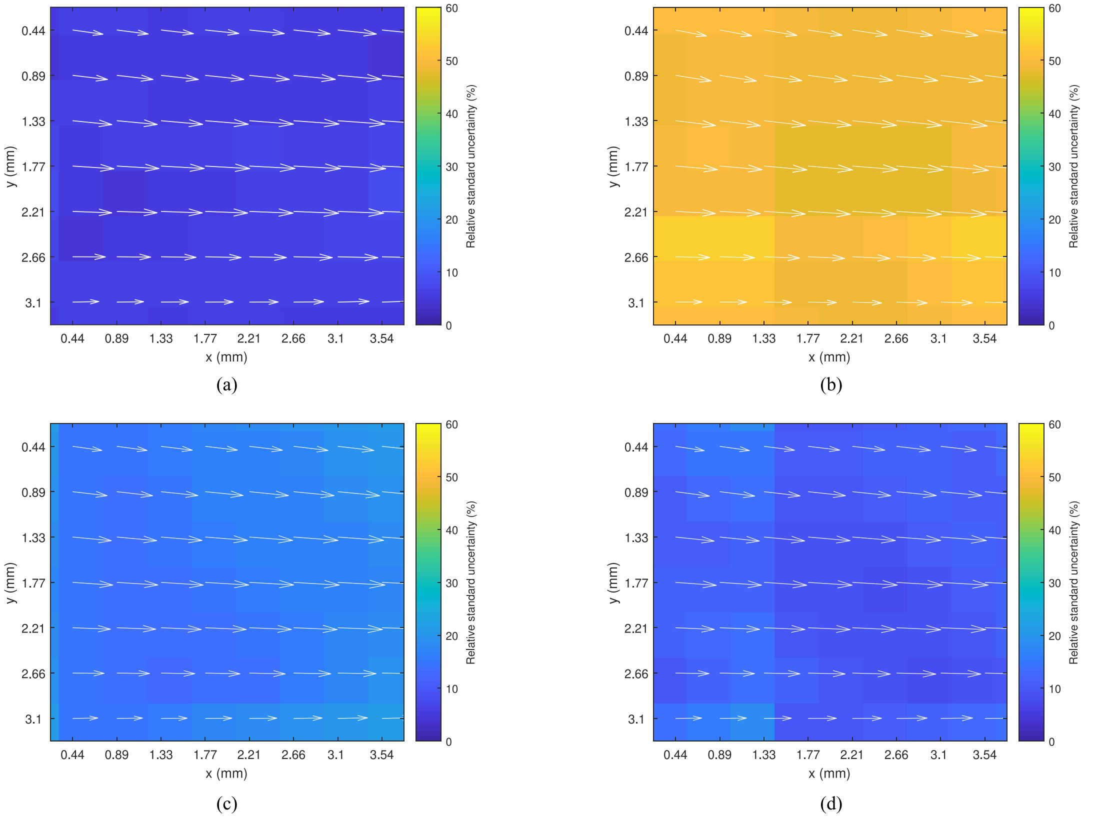
The performance of the adaptive optical correction was characterized in two experiments.

A. Determination of the Attenuation Bandwidth

In the first experiment, the maximum attenuation bandwidth of the system was characterized. Therefore, a defined tilt distortion with a cutoff frequency at 400 Hz was induced by a voice coil mirror (not shown in the optical setup) (see Fig. 9). The plot consists of two graphs, which show the measured Zernike deviation e with the TGS control loop switched ON and OFF. The system is able to attenuate a distortion up to 150 Hz. Further investigation showed that the deformable mirror limits the maximum attenuation bandwidth because the mirror membrane does not follow the Zernike set value correctly anymore above 200 Hz. For example, if a tilt deflection is set to the mirror with a high frequency, the membrane will response in a coma deflection shape. However, the attenuation bandwidth of the reported system is much larger than for other systems reported in the literature, which are in the range of 25 Hz [28] to 43 Hz [1] (ACE system).

B. Flow Velocity Measurement With Time-Varying Distortion

To proof the functionality of the correction system for a relevant application, a PIV flow measurement through a fluctuating air–water interface was conducted. An air flow was directed toward the water surface to excite capillary waves that represent the optical distortion. The pressure of the air flow was chosen low enough that the amplitude range of the deformable mirror



(a)

(b)

(c)

(d)

Fig. 10. (a)–(d) Adaptive optical correction performance of the system. The white arrows indicate the laminar flow field and the background color the local standard deviation. The results are summarized in Table IV. (a) Reference performance of the PIV system, where the control loop and air distortion are switched OFF. The mean relative standard uncertainty is 5.7%. (b) Influence of the air-induced distortion on the PIV measurement with switched OFF control loop. The mean relative standard uncertainty is 49.2%. (c) Reduction of the air induced distortion on the PIV measurement with active Fresnel guide star control loop. The mean relative standard uncertainty is 15.9%. (d) Reduction of the air-induced distortion on the PIV measurement with active transmission guide star control loop. The mean relative standard uncertainty is 11.1%.

was not exceeded. The range of the mirror for a peak-to-valley tip and tilt stroke is 84 μm. This stroke is extended by a Keplerian telescope with a magnification factor of 0.5 to 168 μm. PIV is a method to measure the two-dimensional two-componential velocity flow field in a medium [29]. To visualize the flow, the medium is seeded with reflecting silver-coated hollow glass spheres, and a laser light sheet is used to illuminate the measurement volume. The PIV camera captures two images with a defined time delay $\triangle t$. The images are divided into interrogation areas. For each area, the displacement $\triangle\vec{s}$ of the particles is evaluated by means of a cross-correlation function. With the calculated shift vector and the known time difference $\triangle t$, the local two-component velocity $\vec{v}$ can be calculated with $\vec{v} = \triangle\vec{s} \,/\, \triangle\, t$ for one interrogation area. Repeating this procedure for all interrogation areas yields the two-dimensional flow vector field.

To characterize the improvement of the PIV measurement, the laminar flow behind the nozzle with a mean velocity of about $\overline{v} = 6\ \mathrm{mm} \cdot \mathrm{s}^{-1}$ was measured through the surface for different cases. In Fig. 10, the white arrows indicate the mean flow velocity, and the background color represents the local relative standard uncertainty σ of the flow velocity for 2000 consecutive measurements. The measured standard uncertainty σ is mainly a superposition of three components: the uncertainty of the PIV measurement σ_{PIV}, the instability or turbulence of the flow σ_{flow}, and the uncertainty induced by the fluctuating surface $\sigma_{\mathrm{distortion}}$, which adds a "virtual turbulence" to the measured flow. The relationship between the uncertainties can be expressed as

$$\sigma = \sqrt{\sigma_{\mathrm{flow}}^2 + \sigma_{\mathrm{distortion}}^2 + \sigma_{\mathrm{PIV}}^2}. \tag{7}$$

In Fig. 10(a), the flow was measured with the distortion and control loop switched OFF. This case serves as a reference to determine the lowest reachable relative mean standard uncertainty of about 5.7% for this setup. This value corresponds to the uncertainty of the PIV evaluation σ_{PIV} and a possible small instability of the flow σ_{flow}. If the air flow is switched ON, the mean relative standard uncertainty will increase to 49.2% [see Fig. 10(b)]. The active control loop with the FGS as a feedback value is able to reduce the mean relative standard uncertainty by 67% to 15.9%. With the TGS as a feedback, the relative mean standard uncertainty can be reduced to 11.1%, which is a reduction of 77%.

All cases are summarized in Table IV. The control loop is not able to completely compensate the distortion for both guide stars. This can be explained by noise in the wavefront measurement of the Hartmann–Shack sensor, the adaptive mirror, which is not perfectly imaged onto the water surface, and the correction of

TABLE IV
REDUCTION OF THE STANDARD UNCERTAINTY ACHIEVED BY TGS AND FGS CONTROL LOOPS

	σ $(\mathrm{mm\,s^{-1}})$	$\sigma/\overline{v}$ (%)	Reduction (%)
Reference without disturbance	0.323	5.7	-
Reference with disturbance	2.787	49.2	0
Active control loop with TGS as feedback	0.628	11.1	77
Active control loop with FGS as feedback	0.898	15.9	67

only the first 14 Zernike polynomials. The control loop with the FGS feedback does not reach the value of the TGS loop. This can be explained by uncertainties of the fitted plane to the measured data. Additionally, the concept for the control value recovery will only be valid, if the surface height does not change.

V. Conclusion

FPSoCs are a hybrid solution of a fast native CPU and FPGA on one chip with a shared memory. They are dedicated candidates for realizing a real-time control of complex systems with a short loop time and low latency. In this article, we presented the application of an FPSoC at an adaptive optical wavefront correction system as a case study to correct camera images that were distorted by time-varying aberrations. The system comprised a deformable mirror as an adaptive optical element, a wavefront sensor to determine the distortion, and a control routine realized on the FPSoC. While a short total loop time was achieved by interfacing the camera and the deformable mirror on a low level directly with the FPGA, all computationally nonintensive tasks were implemented on the CPU to keep the flexibility, reusability, and development expense low. With this strategy, the system achieved 3600 control cycles per second with an attenuation bandwidth of 150 Hz of optical distortions of up to 14th-order Zernike polynomial. Camera-based flow velocity measurements were performed using PIV to proof the functionality of the system and to estimate the improvement of the measurement properties achieved by it. In a transmissive approach with two optical access windows, the mean relative velocity standard uncertainty of the PIV flow measurement could be reduced by 77% when the control loop is activated. In a reflective approach using the novel FGS concept, flow measurements can be performed from one side only (single optical access), and the standard uncertainty could be reduced by 67%.

As a perspective, this system can be used to measure the flow inside of droplets with single optical access. This is important in order to improve fuel cells, where water droplets, for instance, block the chemical active membrane. With a more precise understanding of the inner flow field, the water droplets could be removed more effectively with, e.g., a suitable coating [30]. Another example is the investigation of gas flows within Taylor bubbles. Here, a precise understanding of the mass transfer between gas and liquid would enable the development of more efficient and environmentally friendly reactors for the process industry. The authors are convinced that the usage of FPSoCs systems will be a benefit for future industrial applications and industry 4.0, since they close the gap between specialized demanding mission-critical logic implementations and the flexibility and comfort of CPU-based implementations.

References

[1] A. Schimpf, M. Micallef, and J. Charton, “1500 Hz adaptive optics system using commercially available components,” *Proc. SPIE*, vol. 9148, pp. 1662–1669, 2014.

[2] L. F. Rodríguez-Ramos, H. Chulani, Y. Martín, T. Dorta, A. Alonso, and J. J. Fuensalida, “FPGA-based real time controller for high order correction in EDIFISE,” *Proc. SPIE*, vol. 8447, pp. 1021–1029, 2012.

[3] J. Rodriguez-Andina, M. Valdés, and M. J. Moure, “Advanced features and industrial applications of FPGAS—A review,” *IEEE Trans. Ind. Inform.*, vol. 11, no. 4, pp. 853–864, Aug. 2015.

[4] E. Monmasson and M. N. Cirstea, “FPGA design methodology for industrial control systems—A review,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007.

[5] A. Raha, A. Chakrabarty, V. Raghunathan, and G. T. Buzzard, “Embedding approximate nonlinear model predictive control at ultrahigh speed and extremely low power,” *IEEE Trans. Control Syst. Technol.*, to be published, doi: 10.1109/TCST.2019.2898835.

[6] S. Richter, T. Geyer, and M. Morari, “Resource-efficient gradient methods for model predictive pulse pattern control on an FPGA,” *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 828–841, May 2017.

[7] E. N. Hartley, J. L. Jerez, A. Suardi, J. M. Maciejowski, E. C. Kerrigan, and G. A. Constantinides, “Predictive control using an FPGA with application to aircraft control,” *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1006–1017, May 2014.

[8] T. A. Johansen, W. Jackson, R. Schreiber, and P. Tondel, “Hardware synthesis of explicit model predictive controllers,” *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 1, pp. 191–197, Jan. 2007.

[9] J. L. Jerez, G. A. Constantinides, and E. C. Kerrigan, “Towards a fixed point QP solver for predictive control,” in *Proc. IEEE 51st IEEE Conf. Decis. Control*, 2012, pp. 675–680.

[10] Z. Fang, J. E. Carletta, and R. J. Veillette, “A methodology for FPGA-based control implementation,” *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 6, pp. 977–987, Nov. 2005.

[11] C. Hahne, A. Lumsdaine, A. Aggoun, and V. Velisavljevic, “Real-time refocusing using an FPGA-based standard plenoptic camera,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9757–9766, Dec. 2018.

[12] L. Rovere, A. Formentini, and P. Zanchetta, “FPGA implementation of a novel oversampling deadbeat controller for PMSM drives,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3731–3741, May 2019.

[13] Z. Hajduk, B. Trybus, and J. Sadolewski, “Architecture of FPGA embedded multiprocessor programmable controller,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2952–2961, May 2015.

[14] P. Bhatti and B. Hannaford, “Single-chip velocity measurement system for incremental optical encoders,” *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 6, pp. 654–661, Nov. 1997.

[15] G. Lentaris, I. Stratakos, I. Stamoulias, D. Soudris, M. Lourakis, and X. Zabulis, “High-performance vision-based navigation on SoC FPGA for spacecraft proximity operations,” *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2019.2900802.

[16] J. Díaz, E. Ros, F. Pelayo, E. Ortigosa, and S. Mota, “FPGA-based real-time optical-flow system,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, pp. 274–279, Feb. 2006.

[17] H. Kim *et al.*, “A single-chip FPGA holographic video processor,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 3, pp. 2066–2073, Mar. 2019.

[18] V. Galindo *et al.*, “Instabilities and spin-up behaviour of a rotating magnetic field driven flow in a rectangular cavity,” *Phys. Fluids*, vol. 29, no. 11, p. 114104, 2017.

[19] L. F. Rodríguez Ramos, J. J. D. Garcia, J. J. F. Valdivia, H. Chulani, C. Colodro-Conde, and J. M. R. Ramos, “The use of CPU, GPU and FPGA in real-time control of adaptive optics systems,” in *Proc. Adapt. Opt. Extremely Large Telescopes 4, Conf.*, 2015, vol. 1, doi: 10.20353/K3T4CP1131563.

[20] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the Internet of Things and Industry 4.0,” *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[21] R. J. Noll, “Zernike polynomials and atmospheric turbulence,” *J. Opt. Soc. Amer.*, vol. 66, no. 3, pp. 207–211, 1976.

[22] H. Radner, L. Büttner, and J. Czarske, “Interferometric velocity measurements through a fluctuating phase boundary using two fresnel guide stars,” *Opt. Lett.*, vol. 40, no. 16, pp. 3766–3769, 2015.

[23] H. Radner, L. Büttner, and J. Czarske, “Interferometric velocity measurements through a fluctuating interface using a fresnel guide star-based wavefront correction system,” *Opt. Eng.*, vol. 57, 2018, Art. no. 084104.

[24] N. Koukourakis, B. Fregin, J. König, L. Büttner, and J. W. Czarske, “Wavefront shaping for imaging-based flow velocity measurements through distortions using a Fresnel guide star,” *Opt. Express*, vol. 24, no. 19, pp. 22074–22087, 2016.

[25] S. Burgmann, B. Barwari, and U. Janoske, “Oscillation of adhering droplets in shear flow,” in *Proc. 5th Int. Conf. Exp. Fluid Mech.*, Munich, Germany, 2018, p. 500.

[26] P. M. Seiler, M. Gloerfeld, I. V. Roisman, and C. Tropea, “Aerodynamically driven motion of a wall-bounded drop on a smooth solid substrate,” *Phys. Rev. Fluids*, vol. 4, 2019, Art. no. 024001.

[27] G. Guennebaud *et al.*, “Eigen v3,” 2010. [Online]. Available: http://eigen.tuxfamily.org

[28] J. Mocci, M. Quintavalla, C. Trestino, S. Bonora, and R. Muradore, “A multi-platform CPU-based architecture for cost-effective adaptive optics systems,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 10, pp. 4431–4439, Oct. 2018.

[29] M. Raffel, C. E. Willert, F. Scarano, C. J. Kähler, S. T. Wereley, and J. Kompenhans, *Particle Image Velocimetry: A Practical Guide*. Springer, 2018.

[30] S. Milles, M. Soldera, B. Voisiat, and A. F. Lasagni, “Fabrication of superhydrophobic and ice-repellent surfaces on pure aluminium using single and multiscaled periodic textures,” *Sci. Rep.*, vol. 9, no. 1, 2019, Art. no. 13944.

Hannes Radner was born in Rostock, Germany, in 1991. He received the diploma degree in electrical engineering from Technische Universität Dresden, Dresden, Germany, in 2014.

Since 2014, he has been a member of the academic staff of the Laboratory for Measurement and Sensor System Technique, Technische Universität Dresden. He has authored six journal papers. His research interests include laser metrology in combination with wavefront correction systems and the system design and implementation of field-programmable system-on-chip-based control loops for adaptive optics.

Mr. Radner is a recipient of two Gisela-and-Erwin-Sick prizes of Technische Universität Dresden for the best diploma and the best bachelor thesis. He is a Student Member of the International Society for Optics and Photonics.

Johannes Stange was born in Großenhain, Germany, in 1996. Since 2014, he has been studying electrical engineering with Technische Universität Dresden, Dresden, Germany, where he is currently working on his diploma thesis.

His research interests include closed-loop control technology for a wavefront correction system.

Mr. Stange is a recipient of the Gisela-and-Erwin-Sick Prize of the Technische Universität Dresden for the best bachelor thesis. He is a Member of the Deutsche Physikalische Gesellschaft.

Lars Büttner was born in Bad Gandersheim, Germany, in 1972. He studied physics with the Clausthal University of Technology, Clausthal-Zellerfeld, Germany. He received the Ph.D. degree in physics from Leibniz Universität Hannover, Hannover, Germany, in 2004.

From 1999 to 2005, he was a Research Fellow with Laser Zentrum Hannover, Hannover. Since 2005, he has been with the Technische Universität Dresden, Dresden, Germany, where he is currently the Head of the Flow Measurement and Adaptive Laser Systems Group, Laboratory for Measurement and Sensor System Techniques. His current research interests include laser- and ultrasound-based measurement techniques and adaptive optical systems.

Dr. Büttner received an award at the Annual OSA Imaging and Applied Optics Congress for the Second Best Presentation with Focus on Adaptive Optics in 2018. He is a Member of the OSA – The Optical Society, the German Physical Society, and the German Association for Laser Anemometry.

Jürgen Czarske (Senior Member, IEEE) was born in Schleswig-Holstein, Germany, in 1962. He received the Diploma and Ph.D. degrees from the Leibniz University of Hannover, Hannover, Germany, in 1991 and 1995, respectively.

From 1986 to 1991, his research was funded by a scholarship from the company Siemens AG and he worked with the Corporate Technology of Siemens AG, Munich, Germany. From 1991 to 1996, he taught and researched with the Leibniz University of Hannover. From 1995 to 2004, he was responsible for the metrology development with the Laser Center Hannover (LZH e.V.). As a visiting scholar, he conducted research in Japan and the USA from 1996 to 2001. Since 2004, he has been a C4 Professor with the Technische Universität Dresden, Dresden, Germany, where he has been the Director of the Institute of Circuits and Systems since January 2016. He has authored or coauthored around 200 international articles in peer-reviewed journals and held more than 100 invited lectures. He holds more than 20 patents. His research interests include universal control and application of coherent waves using adaptive digital systems. He is known for unconventional optical imaging with wavefront shaping, real-time holography, advanced biomedical imaging, and nonintrusive precision measurements at harsh environments.

Prof. Czarske is a recipient of the 1996 AHMT Measurement Technology Award, the 2008 Berthold Leibinger Innovation Award, the 2014 Reinhart Koselleck Project of the German Research Foundation, and the 2019 Joseph Fraunhofer Prize and the Robert M. Burley Prize of the Optical Society of America (OSA) for research achievements in optical engineering. He is Associate Editor for the *Journal of the European Optical Society—Rapid Publications*, a Supervisor of the Student Chapter Dresden of the International Society for Optics and Photonics (SPIE), an elected member of the Saxon Academy of Sciences in Leipzig, a board member of the German Association for Applied Optics and the German Society for Laser Anemometry, and an elected member of the Scientific Society for Laser Technology (WLT e.V.). He is a Life Fellow of the OSA and the SPIE and a Fellow of the European Optical Society (EOS). He is the General Chair of the General Congress of the International Commission for Optics (ICO) in 2020. The ICO is the umbrella organization for optics and photonics, incorporating academic societies, such as EOS, OSA, IEEE, and SPIE.