Chapter 8

Real-Time Adaptive Optic System Using FPGAs

Steffen Mauch and Johann Reger

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66184

Abstract

For “adaptive optics” (AO) that are used in a control loop, sensing of the wavefront is essential for achieving a good performance. One facet in this context is the delay introduced by the wavefront evaluation. This delay should be kept to a minimum. Since the problem can be split into multiple subproblems, field-programmable gate arrays (FPGAs) may beneficially be employed in view of the FPGAs’ power to compute many tasks in parallel. The evaluation of, e.g., a Shack-Hartmann wavefront sensor (SHWFS) may simply be seen as the evaluation of an image. Therefore, in general, image processing methods may be split into multiple assignments such as connected component labeling (CCL). In this chapter, a new method for real-time evaluation of an SHWFS is introduced. The method is presented in combination with a rapid-control prototyping (RCP) system that is based on real-time Linux operating system.

Keywords: SHWFS, FPGA, real-time Linux, RTAI, CCL, RCP, PCIe, adaptive optics

1. Introduction

“Adaptive optics” (AO) have been successfully utilized for more than one decade to improve the image quality of optical imaging systems. One reason for the high popularity originates from the fact that the image quality may be improved without mechanical adjustment, for example, the lenses. Additionally, the technological progress with respect to the manufacturing of deformable mirrors, an increase of computational power, and new approaches for controlling and sensing the wavefront allows broadening the scope of AO to new application fields, e.g., additive laser manufacturing, general beam shaping, and laser link communication [1].

In **Figure 1**, the general AO principle is illustrated within the context of controlling the wavefront. It is clear that besides good performance with respect to the stroke and the dynamic response of the deformable mirror, the wavefront needs to be measured accurately. To compensate for wavefront distortions, e.g., time-varying disturbances with or without a stochastical and/or dynamical model, the disturbance has to be measured with adequate precision. For the quasi-continuous measurement of the wavefront in AO systems, Shack-Hartmann wavefront sensors (SHWFSs) have widely been employed for measuring the wavefront, thus, the phase of the electromagnetic wave [2–5].

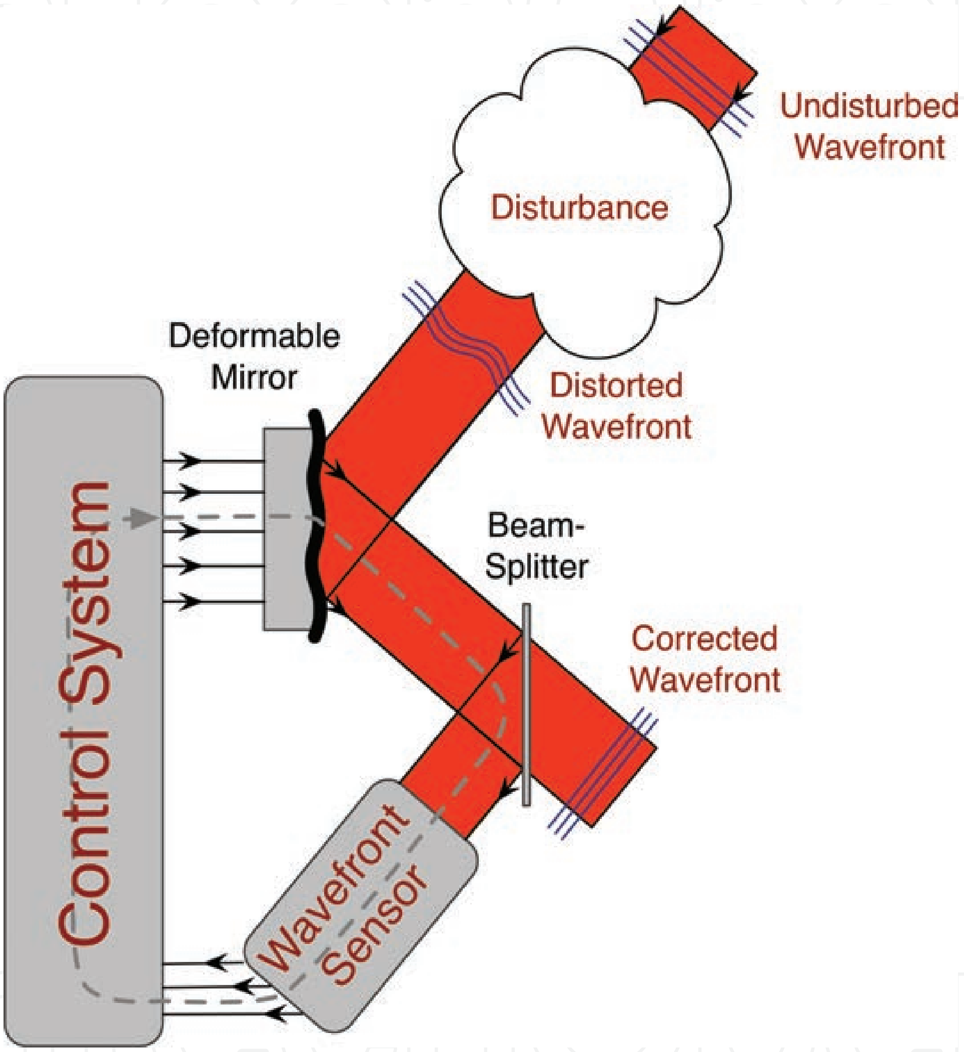


Figure 1. General scheme of adaptive optics, consisting of deformable mirror, wavefront sensor, and control system for closed-loop operation [9].

The SHWFS has shown some performance benefits when compared to interferometers as the SHWFS does not require a reference wave during the measurement process. Furthermore, the measurement sensitivity of an SHWFS primarily depends on the read-out noise of the detector, the luminosity of the wavefront and, hence, the intensity of the spots, and on the algorithm to find and assign the centroids, respectively [6–8].

The most commonly used wavefront measurement sensors, together with their advantages and disadvantages are discussed in Ref. [10]. The SHWFS itself relies decisively on the determination of the centroids, i.e., on the image-processing techniques being applied. The different approaches are elaborated in Ref. [11]. As the computational performance and the dynamic behavior of the deformable mirrors are improving continuously, the sensing of the wavefront should also be accelerated which results in the demand of a low-latency and very large frame-rate. A straightforward attempt is to accelerate the image processing by utilizing parallel approaches; e.g., graphics processing units (GPUs) or field-programmable gate arrays (FPGAs).

The bandwidth demand of closed-loop AO systems is continuously increasing, see Ref. [12] or the report of the European Southern Observatory (ESO) ([13], ch. 7.9), to name but a few. In this regard, the application of GPUs is not as promising as FPGAs for evaluation of the wavefront because the GPU requires the use of the central processing unit (CPU) for data management whereas an FPGA may directly access the image sensor (typically a CMOS or CCD image sensor), that is, the pixel information. This allows parallelism with a low latency and thus a low delay. The problem with the delay is that even just a few milliseconds induced by the wavefront sensor may tend to ruin the overall performance of the closed-loop system as long as no adequate disturbance model is known, see e.g., the Xinetics AO system in Ref. [14]. FPGAs show some flexibility in interfacing to a standard computer, e.g., by using the PCIe interface or Universal Serial Bus 3.0 (USB3.0). Furthermore, the FPGA may be used to perform more tasks, for example, performing the computation for closed-loop operation or interfacing the digital to analog converter (DAC) for controlling the actuators of a deformable mirror without additional expensive cards from the hardware manufacturer.

In the last years, FPGAs became more common in academia but also in the industry due to their enormous capabilities regarding parallelism capability, achievable clocking frequency, and wide logic resources. In this course, FPGAs have been introduced as means for SHWFS evaluation. For instance, in Ref. [15], an FPGA solution is implemented under the assumption that spots cannot leave the associated subapertures.

In this chapter, we present a recently developed rapid-control prototyping (RCP) system that is based on an FPGA, mounted on a hard real-time Linux computer. Using a novel implementation, the evaluation of the SHWFS is performed on the FPGA directly. The implementation guarantees minimum delay during the evaluation of the wavefront and an enhanced dynamic range. We illustrate the algorithm for the spot detection and their ordering. Furthermore, we explain the code generation from a MATLAB/Simulink model to the hard real-time Linux system and the FPGA implementation of the PCIe interface.

2. FPGA-based SHWFS evaluation

For controlling the wavefront in an AO system, the wavefront itself has to be measured in an appropriate way. Several methods have been developed for that purpose, e.g., Pyramid, Shack-Hartmann (SHWFS), Curvature, or Holographic wavefront sensors [3, 16, 17]. Until now, an SHWFS is typically used for this objective as it may offer the best trade-off between performance, flexibility, and price. Since the SHWFS is based on capturing the intensities on an image plane (in general, a complementary metal-oxide semiconductor (CMOS) or charge-coupled device (CCD) image sensor), the evaluation of the SHWFS may be seen as some kind of image processing, calculating image moments. **Figure 2** depicts the basic principle of an SHWFS. Generally, an SHWFS will consist of an array of these lenses, called lenslet array.

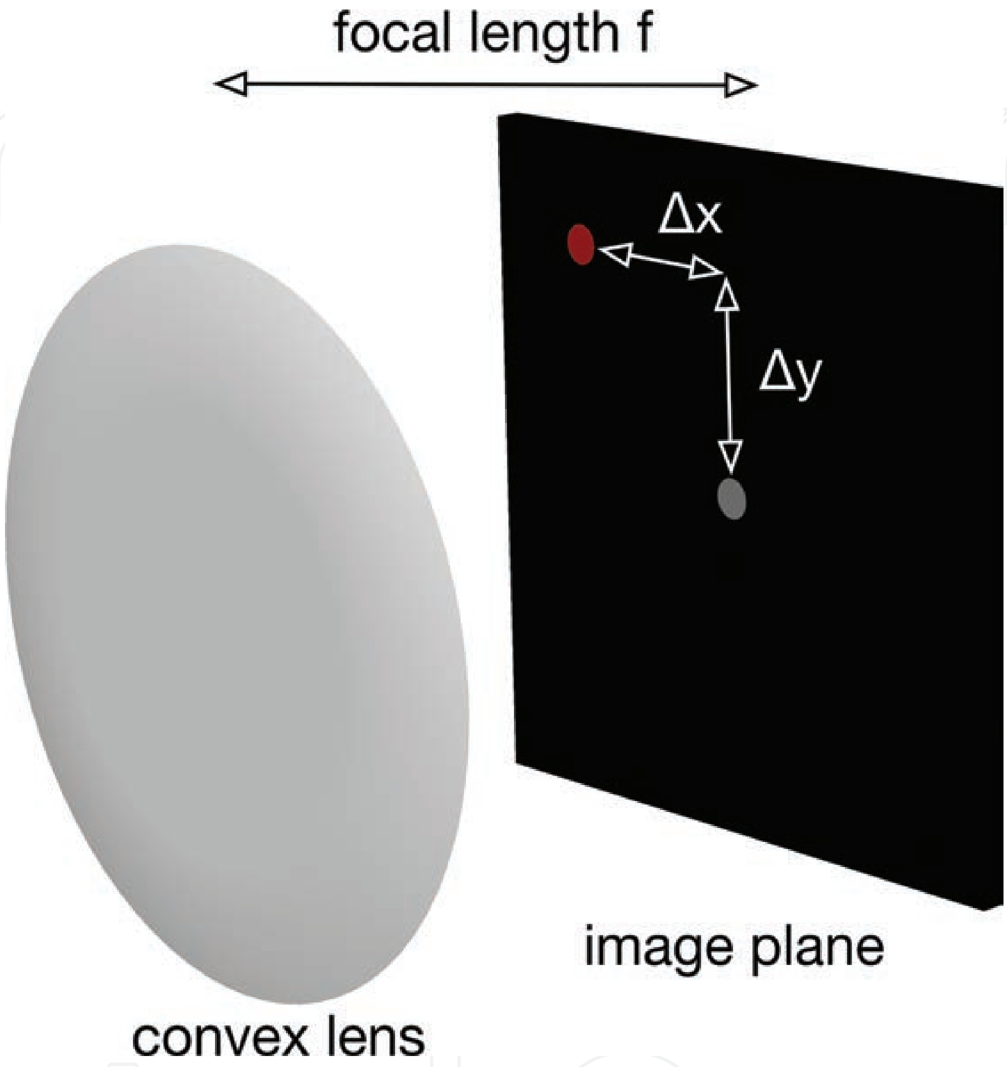


Figure 2. Single convex lens; the gray dot marks the incident flat wavefront, the red dot marks the tilted incident wavefront.

The lenslet array is positioned in parallel to the image plane with a distance of the focal length of the lenses such that the focal point is on the image plane. If a flat wavefront is incident on the lenslet, the spot lies in the projected center of the convex lens in the image plane (marked with the gray dot in **Figure 2**). Due to the nature of the convex lens, the partial derivative of the wavefront with respect to x- and y-direction is averaged over the area of the lens. Thus, the deviation of the focal spot on the image plane with respect to the projected center of the convex lens denotes the local derivative of the wavefront. If a lenslet array is used then one would define a given area on the image plane in which the spot must lie. This may limit, of course, the possible dynamic range because the steepness of the partially tilted wavefront is limited by the area of the image plane and the focal length of the lenslet array.

The task is to determine the position or the deviation Δx and Δy of the spots, as shown in **Figure 2**. For a given area with a predetermined number of pixels, this can be formulated as

$$x_c = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} x_i I(x_i, y_j)}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} I(x_i, y_j)} \tag{1}$$

$$y_c = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} y_i I(x_i, y_j)}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} I(x_i, y_j)} \tag{2}$$

where M and N denote the number of pixel in x- and y-direction, respectively. $I(x_i,y_j)$ is the intensity of the pixel at the coordinate (x_i,y_j).

In the past, several methods have been developed for extending the dynamic range of the SHWFS, such as hardware modification, tracking, similarity approaches, to name but a few. We may now calculate $\Delta x = x_c - X_{pc}$, where X_{pc} is the x-coordinate of the projected center of the convex lens. The corresponding calculation can be done for Δy too. For the work presented here, the SHWFS with its CCD camera is directly connected to an FPGA to the end of lowest possible latency.

Evaluating the pixel information of the SHWFS for determining the phase of the wavefront may be divided into two problems: First, determine the individual centroids, i.e., calculate the centroid of the connected areas and, second, the ordering of the centroids to the lenslet for calculating the deviation with respect to the default position and, thus, computing the local derivatives of the wavefront.

As mentioned previously, as long as a predefined area is given in which the spots have to stay, the dynamic range of the SHWFS is limited. Since the approach of the connected areas does not use any predefined area, the restriction is no longer prevalent. However, to be fair, the default algorithm performs the determination and ordering of the centroids in a single step whereas the ordering is a subsequent step which is discussed in the following.

The determination of the connected areas for calculation of the centroids may be based on different methods. These methods mainly differ in their ability for online calculation of the connected areas, meaning that the pixel stream is processed sequentially at the end of it. Such methods are called single-pass algorithms, emphasizing that only a single pass is required without necessarily storing the complete pixel information. The methods have extensively been studied, e.g., in road sign detection or line tracking systems for lane assistance. The general name for these algorithms is connected-component labeling (CCL). CCL—also denoted by connected-component analysis, region labeling—is an algorithmic application of graph theory. The subsets of connected components are often denoted as "blobs." Blobs are uniquely labeled, based on a predetermined heuristic, mostly along the neighbor relationship.

For this approach, the labeling used for the CCL is based on an eight-point neighborhood system, see **Figure 4**. Another popular neighborhood system is the four-point neighborhood system which is presented in **Figure 3**. In **Figures 3** and **4**, the symbol "*s*" marks the actual pixel and the corresponding neighbors of pixel *s* are marked in gray.

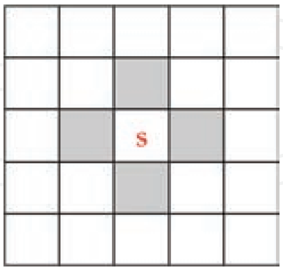


Figure 3. Four-point neighborhood system.

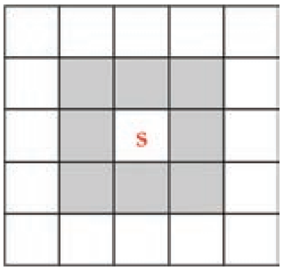


Figure 4. Eight-point neighborhood system.

The procedure for CCL is straightforward. The pixels (the intensity information for each pixel) are streamed sequentially, typically from left to right and top to bottom. If the intensity information is larger than a threshold value, the pixel is assumed to be "1" else "0." This step is called binarization. In **Figure 5**, the boxes with a gray background have already been processed and the actual pixel carries the symbol "?."

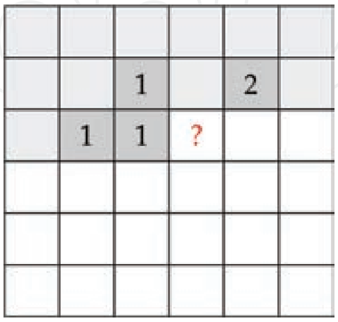
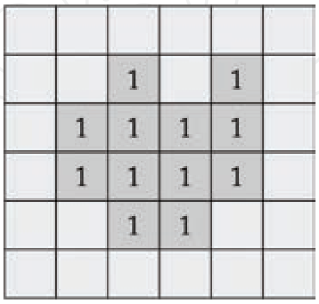


Figure 5. Label collision due to nonconvex blob.

In the case when two sets are connected, but due to the sequential processing receive two different numbers, a label collision may occur. As long as the blobs are convex sets, a label collision cannot occur when using an eight-point neighborhood. Experiments have shown that the assumption of convex blobs is not valid for the typical application scenario of an SHWFS. This may be caused by a disturbed pixel intensity information, recording the noise of the camera sensor, the photon noise, nonperfect lenses, and other effects. Due to thresholding with a fixed value, a single count in terms of the digitalized intensity information can lead to nonconvex blobs, see e.g., **Figure 6**.



		1		1	
	1	1	1	1	
	1	1	1	1	
		1	1		

Figure 6. Resolved label collision due to blob merging.

Application of morphology methods, such as dilation or erosion, is not possible without storing large parts of the image, thus are not single-pass compliant. Additionally, morphology methods significantly increase the delay.

The handling of label collisions can be accomplished by using a label stack which allows label reusing after a label collision has occurred. By means of label reusing the number of provided labels can be kept to a minimum; otherwise, under some circumstances, twice the number or even more labels must be provided. More information is given in Refs. [9, 18, 19].

After having determined the blobs, thus the connected areas, the division of the numerator and denominator may be performed for each valid blob found. The numerator and denominator have to be stored separately as the division step can only be performed when the connected set is maximum. In the block diagram given in **Figure 7**, this step is done in the “centroid calculation/feature extraction” block which also performs the assignment of the centroids to the lenslets.

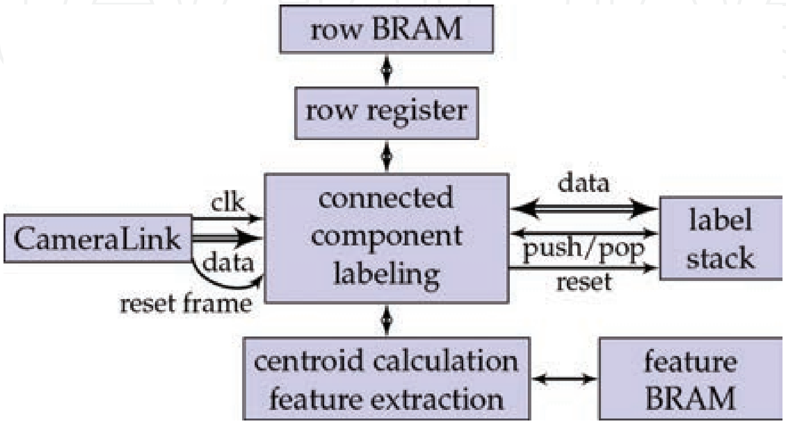


Figure 7. Block diagram of the complete CCL FPGA implementation [18].

One of the key elements of the drafted implementation in **Figure 7** is that solely the former line of the pixel stream has to be stored, not the whole pixel stream. For the applied camera, this results in storing 224 pixels where each pixel is one bit wide because only the binarized value has to be stored. Furthermore, only parts of the former line have to be accessed in parallel such that a small row register is sufficient which is automatically loaded from a Block RAM (BRAM). Using a BRAM has the advantage that the consumption of logic cells is reduced as the BRAM is a dedicated peripheral offered by most FPGAs. The overall logic consumption can be kept at a minimum [18].

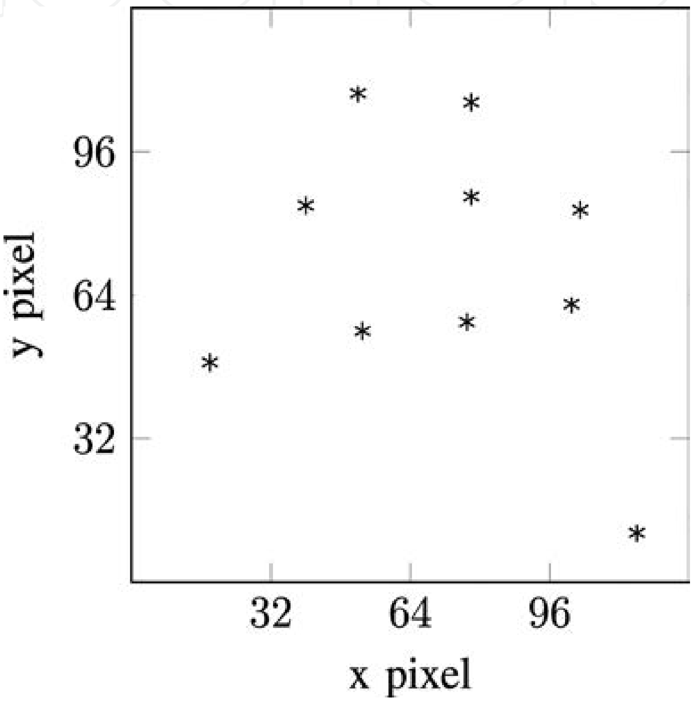


Figure 8. Exemplary centroids for a 4 × 4 lenslet array; some spots are missing.

The assignment or segmentation of the centroids is visualized in **Figures 8** and **9**. This idea has been presented in Ref. [18] and behaves similar to the standard approach for the regular case, that is, the wavefront is not strongly disturbed. However, the advantage of this approach appears whenever a large defocus is present in the wavefront to be measured since shrinking or increasing the overall distance between two neighbored centroids is not a problem for the segmentation method.

The fundamental principle is that the centroids are ordered in parallel with respect to their x- and y-value such that two separately ordered lists exist. Then, straight lines are used to segment the centroids in x- and y-direction by using their distance between each other. As **Figure 9** illustrates, this method is working well also for the case when some centroids are missing due to shadowing or insufficient light intensities. When a very large shearing occurs, however, the method will not be ideal because straight vertical lines are used. But if this problem appears, the standard approach is also not applicable anymore. This algorithm is called simple straight line segmentation.

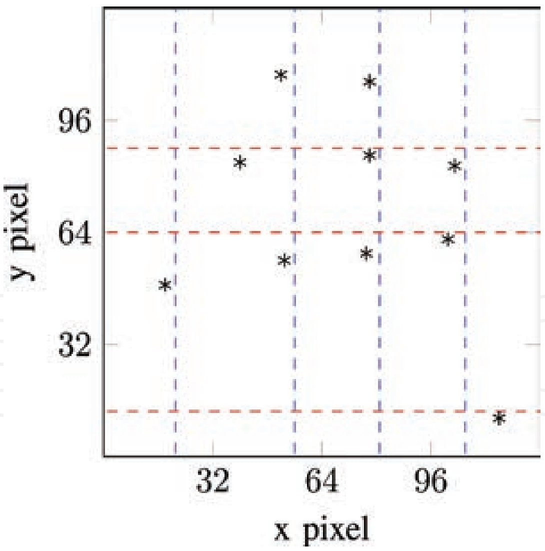


Figure 9. Segmented centroids applying the method presented in [18].

The described algorithm is very simple and straightforward. In Ref. [19] the so-called spiral method has been extended to be deterministic and real-time capable using the centroids gathered by employing CCL. It is obvious that depending on the specific application other methods may be better suited. The CCL may be enhanced by making the thresholding adaptive to compensate the natural intensity inhomogeneity [9, 20]. Another enhancement is the adaptive positioning which for most cases may solve the problem when the number of rows and columns after assignment of the centroids are not the same as with the lenslet array. This circumstance, in general, will lead to ambiguity of the assignment. The adaptive positioning, however, uses an approach based on the similarity of the shape of the segmented centroids and minimizes the shift. Based on this information, the assignment is shifted by one row or column to reduce the offset.

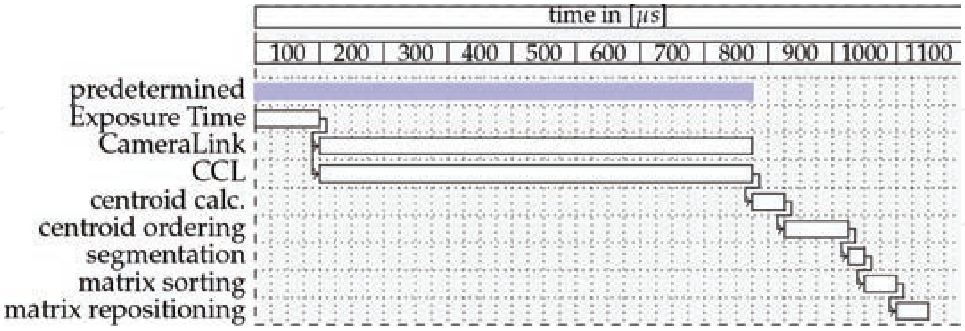


Figure 10. Evolution of processing time, each step rounded up to 25 μs, during SHWFS evaluation applying the FPGA approach.

Figure 10 holds the timeline for the whole evaluation of the SHWFS beginning with the exposure until the assignment of the centroids to the lenslets. The "Imperx ICL-B0620M" camera, on which the "Imagine Optics HASO™ 3 Fast" wavefront sensor is based, is used for this setup. The camera has a maximum frame rate of approximately 900 Hz at 224 × 224 pixel which corresponds to 1111 ms. Thus, the proposed method has a delay equal or less than one single frame.

3. FPGA PCIe integration into the real-time Linux system

The evaluation of the SHWFS is only one part of the AO system since the partial derivatives of the wavefront must be either used for reconstruction of the wavefront and/or used for controlling a deformable mirror (DM) in closed-loop operation. A simple, basic AO concept is used for the work presented in this text, see **Figure 11**. In the experimental setup, the FPGA, besides the evaluation of the SHWFS, is also used for interfacing the digital-to-analog converter (DAC) card. The benefit is that the FPGA can easily guarantee a true parallel output (same guaranteed phase) for all analog outputs even if multiple DACs have to be used.

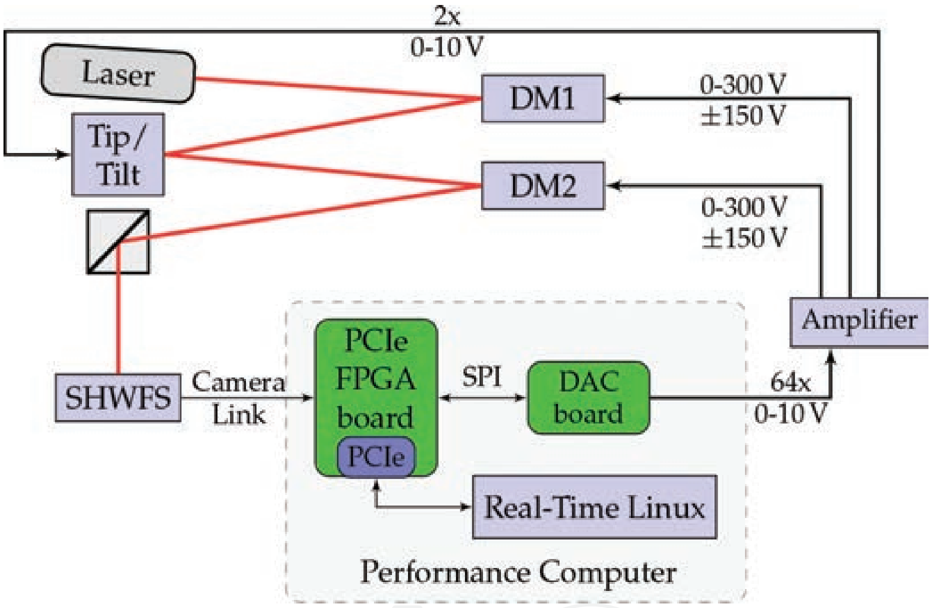


Figure 11. Overview over the basic AO concept; for the detailed concept see [21].

The subsequent processing of the SHWFS data is carried out by a performance computer using state-of-the-art hardware. On this performance computer, the control algorithm is running on a hard real-time Linux operating system (OS). This OS in combination with the performance computer offers rapid-control prototyping (RCP) capabilities in view of the direct MATLAB/Simulink interface. Such an RCP system reduces the implementation effort drastically when different control schemes and approaches need to be tested or compared with each other.

The PCIe FPGA card, see **Figure 12**, is a self-developed card based on the Xilinx Kintex-7 FPGA module TE0741 from Trenz Electronics. The PCIe FPGA card offers more connectivity than only the CameraLink interface. Nevertheless, in this context, only CameraLink, PCIe, and the Serial Peripheral Interface (SPI) are used. The other interfaces are neglected in this context but are presented in detail in Refs. [9, 21].

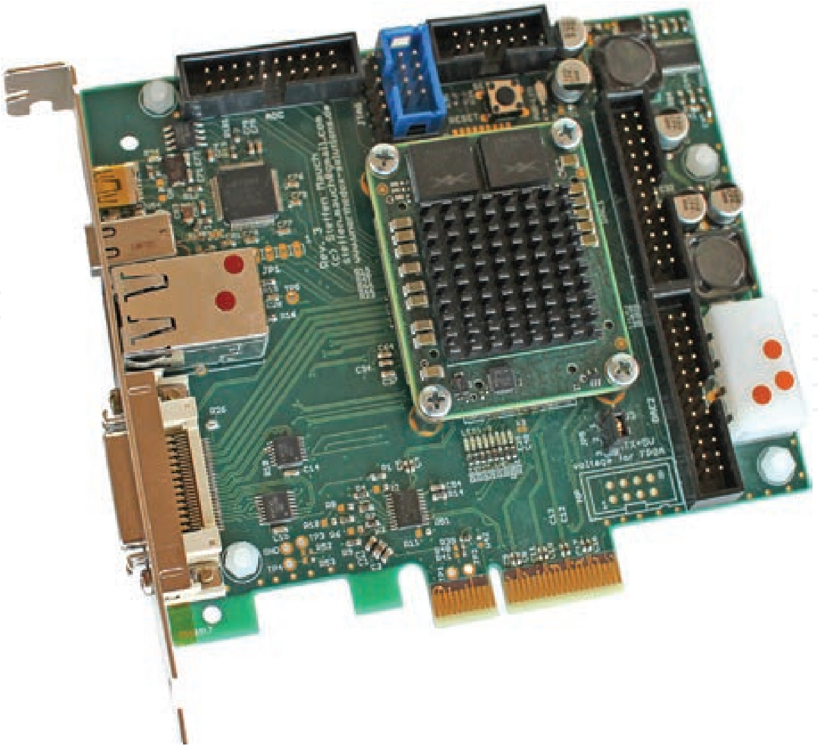


Figure 12. Developed PCIe FPGA board based on a TE0741 (Xilinx Kintex-7) module from Trenz Electronics.

The integration of the FPGA card is realized via the PCIe interface. Thus, almost any modern computer can be used for interfacing the PCIe FPGA card. The SHWFS, more exactly the CCD camera, is connected with the CameraLink interface to the card. Additionally, two separate DAC boards are installed where each DAC board offers 32 analog channels.

The outputs of the DAC cards are fed into an amplifier which amplifies the small signals to drive, for example, the piezoelectric actuators that are part of the DM. In the setup, two DMs have been applied. This circumstance allows the feature that one DM may be used for an artificial, but realistic disturbance generation, whereas the other compensates for such disturbance. In principle, the disturbance may also be virtually induced by adding some signal to the output of the SHWFS; however, a meaningful emulation can be rather involved. This may limit the performance of the system. For this reason, a real disturbance has been incorporated. The amplifier offers the feature to switch between regular and symmetric voltage by modifying the reference ground. Here, the benefit of the symmetric voltage is that the stroke is symmetric as well. Due to the creeping behavior of the piezoelectric actuators, simply applying an offset of [+150] V is not the same as symmetric operation.

For integrating the PCIe FPGA card into the Linux kernel, a kernel driver has to be developed. So as to integrate data acquisition cards, Linux offers a special interface called comedi (control and measurement device interface). Using this interface is very comfortable because the core functionality is already implemented and only low-level driver modules have to be developed for supporting a new data acquisition card. In addition, a user-space library called "comedilib" is available which allows the utilization of user-space to access the functionality of the data acquisition card (**Figure 14**).

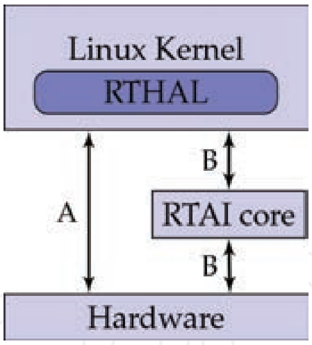


Figure 13. RTAI principle for RTAI-core active or inactive [21].

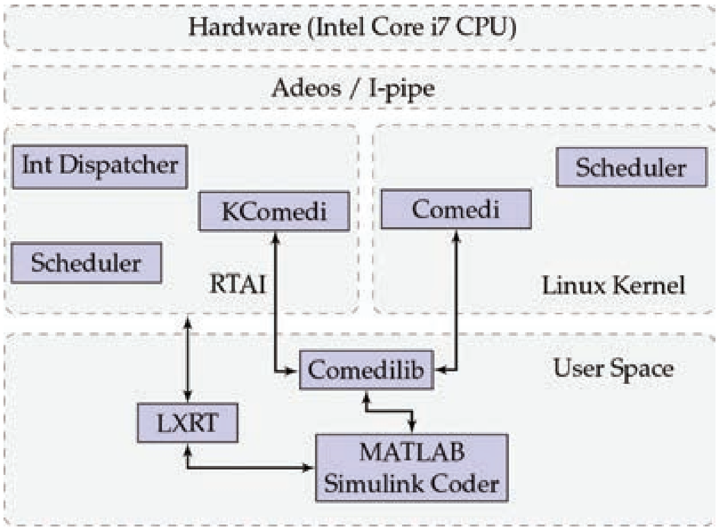


Figure 14. Block diagram of the different abstraction layers used in RTAI/LXRT [9].

The Linux kernel is patched with the RTAI (real-time application interface) [22] patch which itself is based on Adeos. The purpose of the Adeos project is to offer an environment so as to allow sharing of hardware resources among multiple operating systems. RTAI uses that approach (shown in **Figure 13**) for scheduling Linux in the hard real-time support. If RTAI is loaded then case B is active, otherwise case A.

Furthermore, RTAI supports comedi without disturbing the hard real-time behavior. RTAI has the LXRT extension that offers the feature to run real-time applications as user-space programs, see **Figure 14**. Additionally, a MATLAB/Simulink target is available which uses the Simulink Coder for C/C++ code generation [23]. Based on these prerequisites it is easy to extend the given code generation to support more comedi implemented features such as block memory reads or trigger commands.

The PCIe implementation is based on the Xilinx 7 Series Gen2 Integrated Block for PCI Express IP-core which has been extended to support Direct Memory Access (DMA). This way, the FPGA may write the assigned centroids into the main memory of the computer without involving the CPU, see **Figure 15**. PCIe is based on sending and receiving Transaction Layer Packets

(TLPs). The block "COMEDI_SHWFS_READ," see **Figure 16**, performs a blocking read request on the memory destination also being used for the DMA transfer. Behind these Simulink blocks, we have predefined *s*-functions which are based on the functionality provided by comedi. The "COMEDI_SHWFS_TRIGGER" triggers the start of the frame capture; thus, the image acquisition is synchronous to the real-time application which is essential for guaranteeing a deterministic behavior. As shown in the timeline in **Figure 10**, after approximately 1050 μs the data is transferred via DMA to the main memory of the computer.

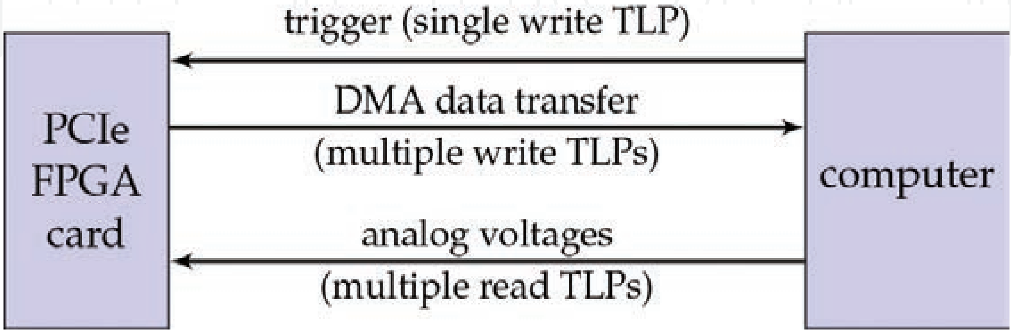


Figure 15. Communication using PCIe interface between FPGA and computer.

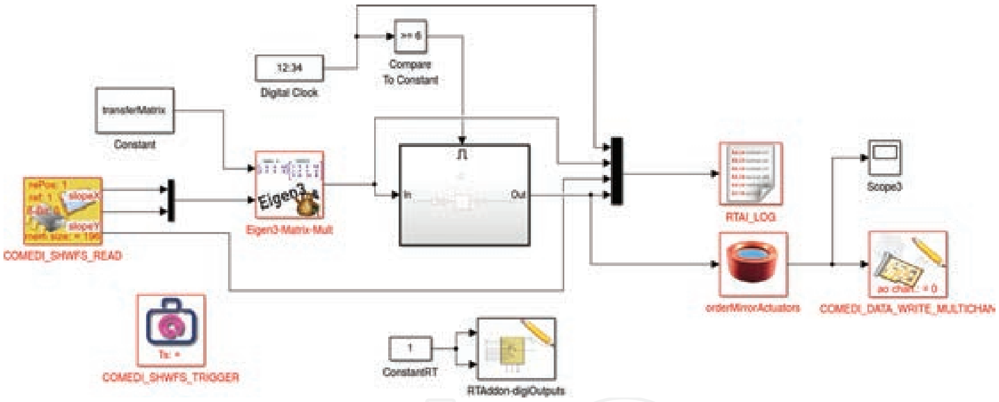


Figure 16. Simulink model used for code-generation and based on Simulink Coder.

The captured data, e.g., from the SHWFS as well as the control output and error values are fed into the "RTAI_LOG" block. This module creates an interface with which another user-space program may record the data and write it either to the main memory or the hard disk.

4. Results from the adaptive optics setup

For closing the loop of the AO setup, a stabilizing controller is required. In Ref. [9], an $\mathcal{H}_\infty$ controller has been synthesized which robustly stabilizes the AO system. In the past, in general, PI(D) controllers have been used which often were tuned by hand. It would go too far to present the method for controller synthesis. So, this is not exposed in this survey. Since the AO setup uses an RCP approach, it is not very time consuming to test other control schemes, as only the Simulink model has to be adjusted accordingly. Of course, the design of a stabilizing controller which is also robust against a set of uncertainties may be rather complicated and time consuming.

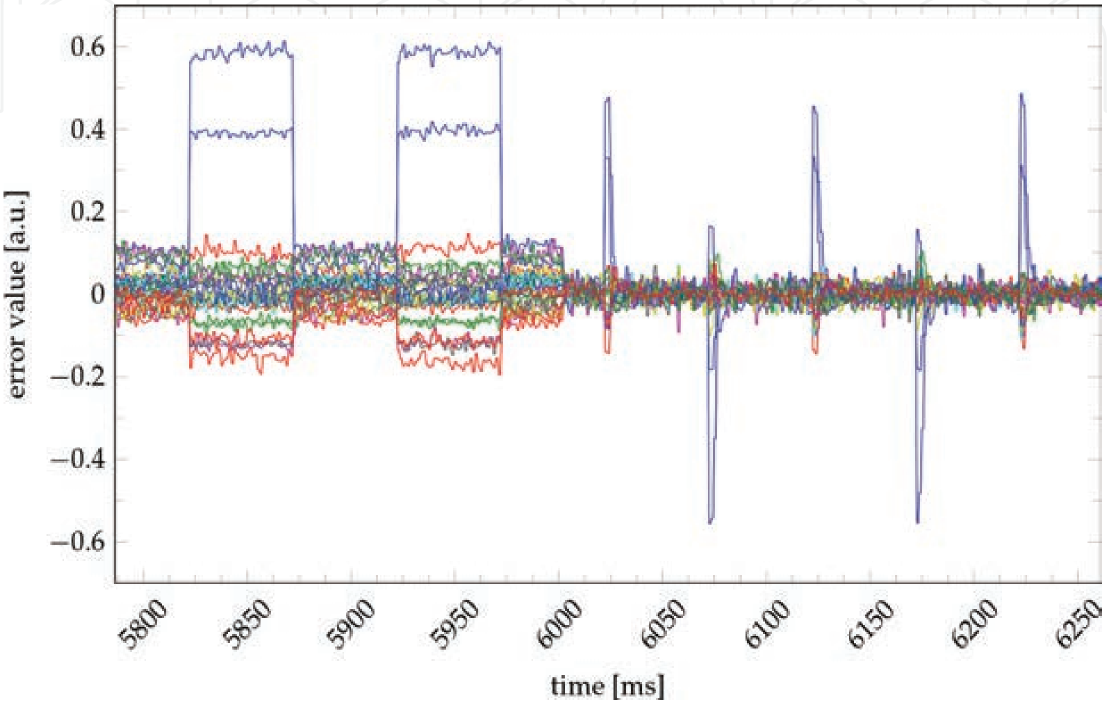


Figure 17. Application of a 10 Hz step disturbance while controller is switched on at $t = 6\,\mathrm{s}$; $f_{\mathrm{controller}} = 1600\,\mathrm{Hz}$ and $f_{\mathrm{shwfs}} = 800\,\mathrm{Hz}$ [9].

To validate the applicability of the presented approaches, **Figure 17** shows some recorded data. The controller has been switched on at time instance 6000 ms. The disturbance is a 10 Hz rectangular offset that is applied to one actuator of DM1. As the actuator patterns of the DM2 and DM1 are not the same and, additionally, do not have the same number of actuators, the result is that multiple actuators are required for compensating the disturbance. A rectangular disturbance is ideally suited to visualize the power of the controller as the steady-state error as well as the time required for compensating the disturbance may be analyzed.

The error value is obtained after the multiplication of the control matrix with the centroids (in **Figure 16** the signal after the "Eigen3-Matrix-Mult" block). The control matrix itself is the pseudo-inverse of the actuator influence function [8, 9].

The dimension of the error value is the same as the number of actuators. Nevertheless, the error value itself does not give a direct insight on how the wavefront looks like. Therefore, **Figure 16** visualizes the reconstructed wavefront as a 3D surface. The respective error values are depicted in **Figure 18** separately. To calculate the Strehl values based on the reconstructed wavefront in **Figure 20**, the wavefront at time instance 6320.63 ms has been used as reference;

thus, showing a Strehl value of exactly one. Both the three-dimensional representation and the error value visualize that it took 3–4 frames to reject the disturbance.

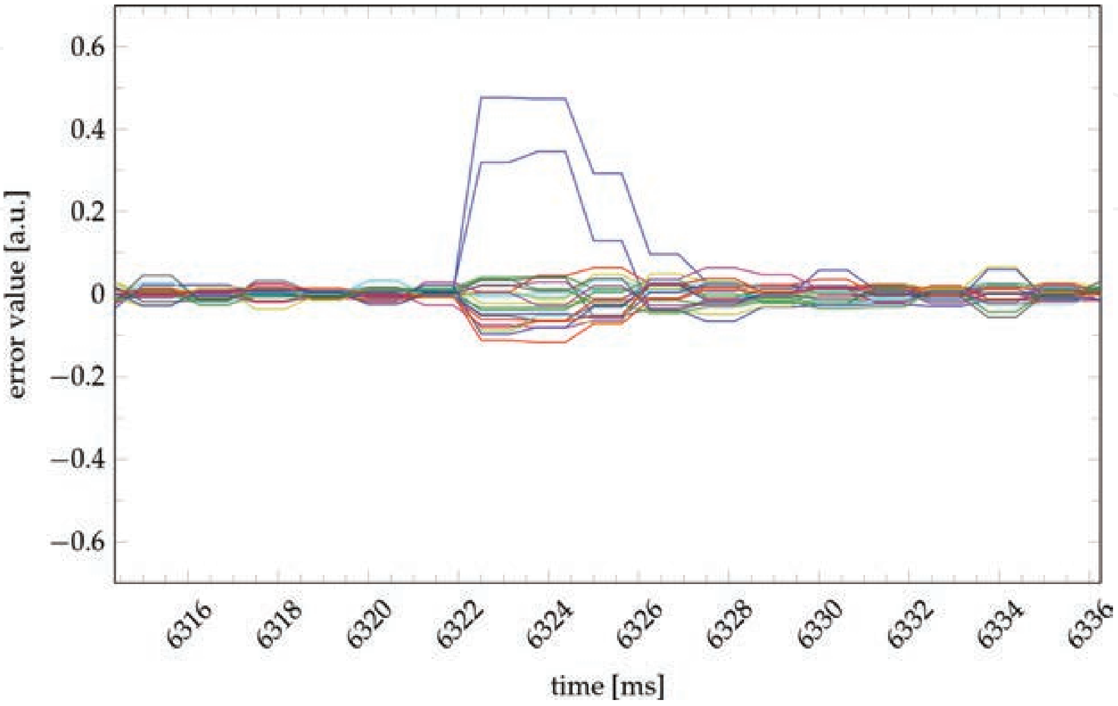


Figure 18. Experimental data with zoomed *x*-axis to highlight the control behavior after a step disturbance, $f_{\text{controller}} = 1600\text{Hz}$ and $f_{\text{shwfs}} = 800\text{Hz}$ [9].

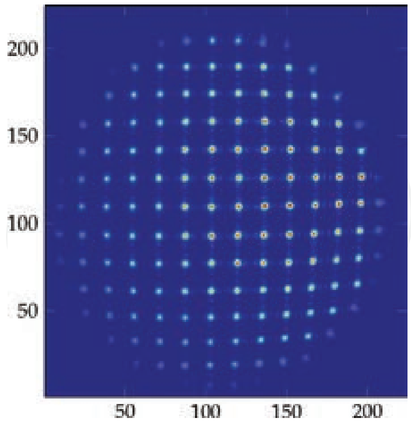
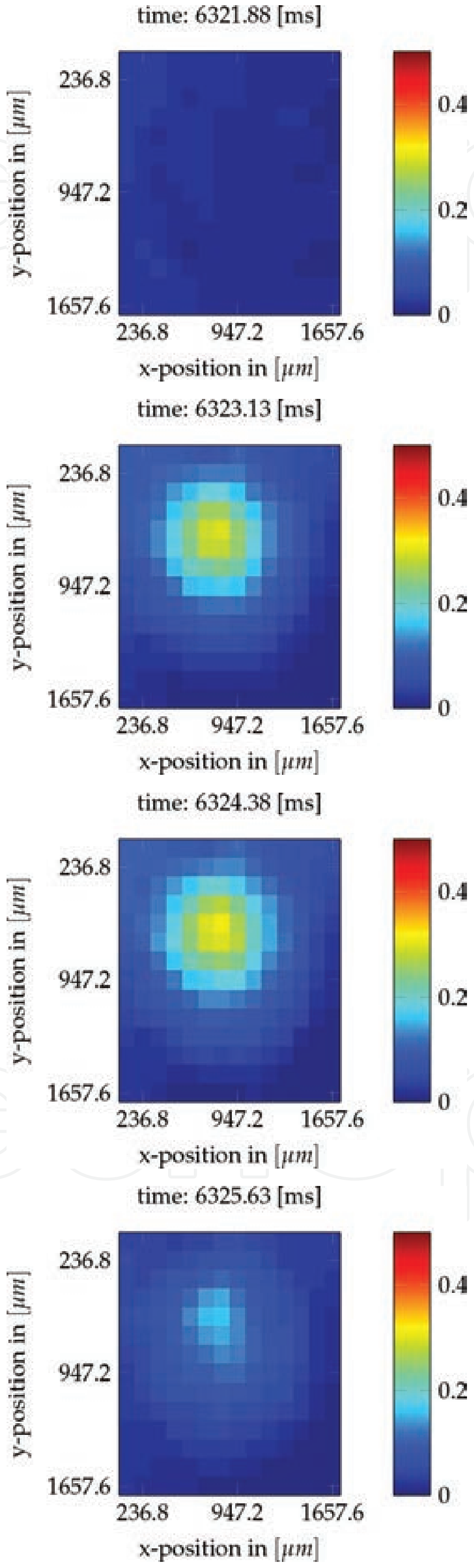


Figure 19. Captured image of the SHWFS, having a lenslet array of 14×14, during experiments.

Finally, **Figure 19** shows some captured camera image of the SHWFS. The colors have been adjusted for better visualization. The SHWFS has a lenslet array of 14 × 14 while the pixel area is 224 × 224 pixels. The standard approach cannot be used here as the spots are leaving the area on the image sensor. The area on the image sensor for each lens would be 16 × 16 pixels.

However, the new approach has no problem and correctly assigns the spots to the lenses and thus correctly measures the wavefront (**Figure 20**).



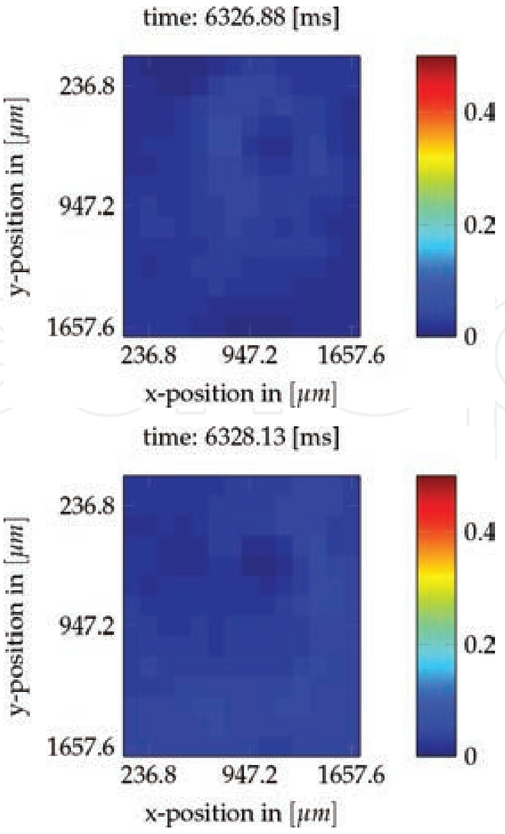


Figure 20. Reconstructed wavefront, rejecting a step disturbance; same data as in **Figure 18** [9].

5. Conclusion

The use of FPGAs in the context of AO has proven to be very beneficial with respect to the achievable performance, especially in closed-loop operation. One positive aspect is the direct evaluation of the SHWFS on the FPGA which allows to minimize the delay and to increase the throughput. For the evaluation of the SHWFS, new approaches have been presented which surpass or considerably extend existing methods. However, they have not reached possible limits so far, particularly, in terms of the achievable dynamic range of the SHWFS.

The practical applicability of the method has been demonstrated in various experiments paired with extensions such as the adaptive repositioning and thresholding [9, 18–20].

For designing AO setups and optimizing its performance, interdisciplinary groups are indispensable. In this context, the control engineers may synthesize their simulation models directly in code for closed-loop operation. Such an RCP system may also be a commercial solution such as dSpace. Yet, the presented RTAI-based hard real-time Linux system has the important benefit to be of far lower initial cost with respect to hardware while granting higher flexibility and ease of customization.

Author details

Steffen Mauch[1*] and Johann Reger[2]

*Address all correspondence to: steffen.mauch@tu-ilmenau.de

1 Ingenieurbüro Mauch, Unorthodox Solutions, Germany

2 Technische Universität Ilmenau, Ilmenau, Germany

References

[1] M. Knapek. Adaptive Optics for the Mitigation of Atmospheric Effects in Laser Satellite-To-Ground Communications. *Dissertation, Technische Universität Munchen*, 2011.

[2] P.L. Wizinowich. Adaptive optics and keck observatory. *IEEE Instrumentation Measurement Magazine*, 8(2):12–19, 2005. ISSN 1094-6969. doi: 10.1109/MIM.2005. 1405918.

[3] R.K. Tyson and B.W. Frazier. *Field Guide to Adaptive Optics, 2nd Ed (SPIE Field Guide Vol. FG24).* SPIE Press, 4, 2012. ISBN 9780819490179.

[4] L.C. Andrews. *Field Guide to Atmospheric Optics (SPIE Vol. FG02).* SPIE Publications, 1, 2004. ISBN 9780819453181.

[5] P. Hickson. *Fundamentals of Atmospheric and Adaptive Optics*. Homepage University of British Columbia, 2008.

[6] R. Irwan and R. G. Lane. Analysis of optimal centroid estimation applied to Shack-Hartmann sensing. *Applied Optics*, 38(32):6737–6743, 1999. doi: 10.1364/AO.38.006737.

[7] Z. Jiang, S. Gong, and Y. Dai. Monte-Carlo analysis of centroid detected accuracy for wavefront sensor. *Optics & Laser Technology*, 37(7):541–546, 2005. doi: 10.1016/j.optlastec. 2004.08.009.

[8] R.K. Tyson. *Principles of Adaptive Optics, Fourth Edition.* CRC Press}, ISBN: 9781482252330, 11, 2015.

[9] S. Mauch. *Robust Control of an Adaptive Optics System*. Dissertation, Technische Universität Ilmenau, 2016.

[10] O. Guyon. Limits of adaptive optics for high-contrast imaging. *The Astrophysical Journal*, 629(1):592, 2005. doi: 10.1117/12.789849.

[11] A.M. Nightingale and S. Gordeyev. Shack-Hartmann wavefront sensor image analysis: a comparison of centroiding methods and image-processing techniques. *Optical Engineering*, 52(7):071413–071413, 2013. doi: 10.1117/1.OE.52.7.071413.

[12] A. Basden, D. Geng, R. Myers, and E. Younger. Durham adaptive optics real-time controller. *Applied Optics*, 49(32):6354–6363, 2010. doi: 10.1364/AO.49.006354.

[13] K. Kepa, D. Coburn, J.C. Dainty, and F. Morgan. Re-baselining the ESO ELT project. *Analysis and Roadmap from the ELT Adaptive Optics Working Group*, 2006.

[14] R.M. Rennie, D.A. Duffin, and E.J. Jumper. Characterization and aero-optic correction of a forced two-dimensional weakly compressible shear layer. *AIAA Journal*, 46(11): 2787–2795, 2008. doi: 10.2514/1.35290.

[15] K. Kepa, D. Coburn, J.C. Dainty, and F. Morgan. High speed optical wavefront sensing with low cost FPGAs. *Measurement Science Review*, 8:87–93, 2008. doi: 10.2478/ v10048-008-0021-z.

[16] J.M. Geary. *Introduction to Wavefront Sensors (Tutorial Texts in Optical Engineering).* Society of Photo Optical, 5, 1995. ISBN 9780819417015.

[17] U. Schnars, C. Falldorf, J. Watson, and W. Jüptner. *Digital Holography and Wavefront Sensing: Principles, Techniques and Applications.* Springer, 2nd ed., 2015 edition, 7, 2014. ISBN 9783662446928.

[18] S. Mauch and J. Reger. Real-time spot detection and ordering for a Shack-Hartmann wavefront sensor with a low-cost FPGA. *IEEE Transactions on Instrumentation and Measurement*, 63(10):2379–2386, 2014. ISSN 0018-9456. doi: 10.1109/TIM.2014.2310616.

[19] S. Mauch and J. Reger. Real-time implementation of the spiral algorithm for Shack-Hartmann wavefront sensor pattern sorting on an FPGA. *Measurement*, 92:63–69, 2016. ISSN 0263-2241. doi: 10.1016/j.measurement.2016.06.004.

[20] S. Mauch, A. Barth, J. Reger, N. Leonhard, and C. Reinlein. Improved thresholding and ordering for Shack-Hartmann wavefront sensors implemented on an FPGA. *International Workshop on Adaptive Optics for Industry and Medicine (AOIM)*, 2015.

[21] S. Mauch, J. Reger, C. Reinlein, M. Appelfelder, M. Goy, E. Beckert, and T. Tünnermann. FPGA-accelerated adaptive optics wavefront control. *Proceedings of SPIE*, 8978:1–12, 2014. doi: 10.1117/12.2038910.

[22] P. Mantegazza, E. L. Dozio, and S. Papacharalambous. Rtai: Real time application interface. *Linux Journal*, 2000(72es), April 2000. ISSN 1075-3583.

[23] R. Bucher and S. Balemi. Rapid controller prototyping with matlab/simulink and linux. *Control Engineering Practice*, 14(2):185–192, 2006. ISSN 0967-0661. doi: 10.1016/j.conengprac.2004.09.009. Special Section on Advances in Control Education Symposium.