FPGA Implementation of a Kalman-Based Motion Estimator for Levitated Nanoparticles

Jiawei Liao®, Michael Jost, Michael Schaffner, Michele Magno, Senior Member, IEEE, Matthias Korb, Member, IEEE, Luca Benini, Fellow, IEEE, Felix Tebbenjohanns, Rene Reimann, Vijay Jain, Michael Gross, Andrei Militaru, Martin Frimmer, and Lukas Novotny

Abstract—Optically trapped nanoparticles are used in various fields ranging from biophysics to precision sensing. An optically trapped nanoparticle can be regarded as a harmonic oscillator driven by the thermal fluctuations of its environment. Unlocking the potential of optically levitated systems for precision measurements in the classical and the quantum regime requires cooling of the particle motion. In parametric feedback cooling, the center-of-mass motion of a nanoparticle optically levitated in a vacuum is reduced by temporally modulating the optical trapping potential. This technique requires a precise measurement of the particle’s motion to derive the feedback signal and is prone to measurement noise and inevitable thermal process noise. In a state-of-the-art implementation, the feedback signal is derived from a simple phase-locked loop (PLL). Kalman filters are regularly deployed in a variety of application scenarios to improve system performance under noisy conditions. In this paper, we investigate theoretically and experimentally the performance of parametric feedback cooling, where the measurement signal is Kalman filtered before entering the PLL. Compared to sole PLL cooling, our numerical full-system simulations show a 20% reduction of the residual motional energy of a trapped nanoparticle in the presence of the Kalman filter. We detail a field-programmable gate array-based implementation of a Kalman filter and evaluate its performance in-field.

Index Terms—Field-programmable gate array (FPGA), Kalman Filter, nanophotonics, parametric cooling, signal processing, state estimation.

I. INTRODUCTION

IN RECENT years, nanomechanical oscillators have become important tools for precision measurements [1], [2]. A particularly intriguing realization of a nanomechanical oscillator is a nanoparticle levitated in a vacuum by the forces generated by a strongly focused laser beam. Since the optically

Fig. 1. High-level overview of the optical trapping and cooling experiment [11]. Inside a vacuum chamber, a spherical dielectric nanoparticle (diameter 134 nm) is trapped in a strongly focused laser beam. The particle’s position along the three spatial directions x, y, and z is measured optically using photodetectors. From the measured position, a feedback signal is derived using a PLL. This feedback signal is applied to an EOM, which modulates the intensity of the trapping laser. The resulting modulation of the optical trap stiffness damps the particle motion and cools its center-of-mass temperature to the mK range. levitated nanoparticle does not suffer from clamping losses, it resembles a strongly underdamped harmonic oscillator with an exceptionally high-quality factor [3]. For this reason, levitated nanoparticles have been utilized for ultrasensitive force measurements [4]–[6], and the production and sensing of mesoscopic mechanical quantum states may be within reach [7]–[9].

A host of exciting proposals relies on removing all thermal energy from a levitated particle (i.e., cool its center-of-mass motion), and bringing it to the quantum ground state of motion [8]–[10]. The most widespread approach in this direction is parametric feedback cooling of the particle center-of-mass motion. In this technique, schematically illustrated in Fig. 1, the intensity of the trapping laser beam is modulated in order to slow down the motion of the trapped particle [3]. Parametric feedback cooling relies on a precise measurement of the particle’s trajectory to derive the feedback signal. In a standard implementation, the feedback signal is derived from a phase-locked loop (PLL) locking to the particle’s oscillation in the trap [11]. The performance of this setup mainly depends on the PLL’s ability to track the phase and frequency of the position signal correctly. Two noise sources in the system make this task difficult: measurement noise and process noise. Importantly, as parametric cooling reduces the particle’s amplitude, the relative impact of the noise increases as the particle’s motion is cooled and the task of the PLL becomes increasingly difficult. At some point, the PLL is unable to track the
position signal. This point marks the performance limit of the cooling system.

Since the system to be controlled is conceptually very simple (a harmonic oscillator), it is an intriguing idea to deploy a Kalman filter to strip the noise off the measurement record of the particle motion [12]–[14]. Kalman filters are extensively used in a wide range of applications [15]–[18] to filter noisy data. While feedback cooling of a levitated particle based on a Kalman-filtered measurement of its motion has been demonstrated [14], a quantitative analysis of the benefits and drawbacks of Kalman filtering in the context of feedback cooling is still outstanding. In particular, in the context of parametric feedback cooling of nanomechanical oscillators, it is an open question which performance increase can be expected from a Kalman filter. Furthermore, a quantitative in-field investigation of such a system has remained elusive to date.

The main contributions of this paper are as follows.

1) We deploy a Kalman filter to process the position measurement record of an optically levitated silica nanoparticle (nominal diameter 136 nm) before feeding that record into a PLL to derive the feedback signal for cooling the particle's motion. The Kalman filter computes an optimal estimate of the particle position based on a physical system model and the (noisy) position measurements.

2) We present numerical full system simulations indicating a 20% increase in cooling performance utilizing a Kalman filter as compared to the standard approach based solely on a PLL.

3) We present an accurate field-programmable gate array (FPGA)-based embedded Kalman filter system and provide key insights from in-field evaluations. In particular, we verify that this system is functional, and we perform initial comparisons between cooling with and without Kalman filter. These experiments reveal that careful tuning of the Kalman filter parameters at runtime is crucial for a good performance.

In our current setup, we find that the system with Kalman filter yields similar cooling performance compared to the system without it, and we note that the predicted improvement has not yet been achieved experimentally. To this end, we plan to optimize the experimental arrangement and investigate the possibility of tuning the Kalman filter parameters dynamically.

II. PRELIMINARIES

A. Related Work

Real-time signal processing using Kalman filters is computationally intensive, which makes it challenging to reach sufficiently high sampling rates and low jitter when implemented on processor-based systems [19], [20]. To this end, modern FPGAs provide a cost-effective and rugged alternative, since they have over the past years become powerful enough to implement complex signal processing circuits like Kalman filters [20]. Hence, FPGA-based Kalman filter implementations have been recently investigated in many application domains, such as antilock braking systems [19], interferometry [21], sensors systems [16] and radar tracking systems [22]. In the domain of nanophotonics, we are only aware of two previous studies by Jost et al. [13]. This paper is an extension of the work by Jost et al., where preliminary results based on simulation have been presented. In particular, the validation part of the system has been extended, experiments and comparisons with nanoparticles at low pressure have been added, and the experimental difficulties encountered are discussed. Setter et al. [14] present a similar system arrangement based on two Virtex-5 FPGA—one for the Kalman filter, and one for frequency doubling and phase shifts that are implemented with an external PLL in our setup (Fig. 2). Their implementation uses fixed-point arithmetic and reaches a sampling rate of 440 kHz, whereas our design employs single-precision floating-point (FP) for maximum precision at a sampling rate of 500 kHz. In addition, we also provide a full system simulation to estimate the maximum achievable cooling performance improvement over the established PLL-based approach and perform initial experimental comparisons.

B. Experimental System

Fig. 1 illustrates our experimental setup [3], [11]. A laser beam (wavelength 1064 nm) is focused on a tight spot using a microscope objective (numerical aperture 0.9NA, focal power 100 mW) in the vacuum chamber. A second lens (0.77NA) collects the outgoing light, which is sent through a waveplate (λ/2) and a polarizing beam splitter cube. Finally, the light passes through an optical isolator and is detected with balanced photodetectors. The detected signal is proportional to the particle position along the x-direction and is sent to the electronics deriving the feedback signal. The feedback electronics are either a standard PLL, or a Kalman filter followed by the standard PLL. The feedback signal is applied to an electrooptic modulator (EOM) modulating the intensity of the trapping laser. Since the oscillation frequencies for the three degrees of freedom of the particle (x, y, and z) are different, the feedback signals derived for the three axes can be superposed and intensity modulation of the single trapping laser provides a means to cool all three degrees of freedom simultaneously.
C. Parametric Feedback Cooling

Parametric feedback cooling modulates the stiffness of the optical trap to dampen the motion of the nanoparticle. The state-of-the-art approach to parametric feedback cooling is based on a PLL locked to a measurement of the position \( x(t) \) of the particle. We detail the principle of parametric feedback cooling for the \( x \) motion of the particle. Our discussion naturally extends to the \( y \) and \( z \) degrees of freedom. Using parametric feedback, cooling of a levitated nanoparticle to a center-of-mass temperature \( T_{\text{c.m.}} \sim 450 \) \( \mu \)K has been demonstrated [11]. The principle of parametric feedback cooling is schematically illustrated in Fig. 3. Whenever the particle moves away from the trap center, the trap stiffness is increased by increasing the power of the trapping laser. In contrast, whenever the particle moves toward the trap center, the trap stiffness is reduced. Accordingly, the trap stiffness is modulated periodically at twice the oscillation frequency of the particle. Parametric feedback cooling can be applied to all degrees of freedom of the particle simultaneously if their eigenfrequencies are sufficiently different.

Clearly, for the feedback system to cool the particle’s motion, it is critical that the feedback signal generated by the PLL is locked both in frequency and in phase to the measurement signal \( x(t) \). Importantly, two factors limit the quality of the PLL output. First, the measurement record of the particle position is obscured by measurement noise. Since the particle position is measured optically in the system under consideration, the fundamental limit of this measurement noise is photon shot noise. Second, collisions of the levitated particle with residual gas molecules in the vacuum chamber, and at very low gas pressures momentum transfer from photons [11], generate a random force that plays the role of process noise. This is where the Kalman filter approach comes into play. At the heart of our work stands the idea to first track the particle’s position by means of a Kalman filter to then feed the Kalman-filtered signal into the PLL. The Kalman filter optimally estimates the state of a linear system if all system noise is white and Gaussian. The system considered is linear, and the dominating noise sources can be well approximated by white Gaussian noise over the bandwidth of interest. This is due to the fact that the photodetector integrates the responses of a large number of photons (around \( 5 \times 10^{11} \) over an integration time of 1 \( \mu \)s). The photon-shot noise follows a Poissonian distribution, and hence, the photodetector output can be seen as the sum of a large number of independent, Poissonian distributed events. Due to the central limit theorem, this sum can be well represented by a Gaussian distribution. This fact has been verified experimentally, as described in the caption of Fig. 4. Hence, our system can be expected to work well in practice, since the Kalman filter can show good results even for not perfectly white and Gaussian noise [23].

III. Evaluation of Parameters and Performance

To assess the feasibility of the Kalman filter approach and to evaluate important system parameters like bit widths and sampling frequencies, we set up a closed-loop full system simulation for one axis of the physical particle system. We employ MATLAB Simulink to model the system. The top-level schematic is shown in Fig. 5. The Physical System block models the particles motion in time and space. The particles position \( x(t) \) is measured which results in the signal \( \hat{x}(t) \). This analog signal is converted to the digital signal \( \hat{x}_k \), which is input to the Estimator block that implements the Kalman filter. The position estimate \( \hat{x}_k \) is then fed into the digital-to-analog converter (DAC) that is followed by a Reconstruction Filter. The reconstructed signal \( \hat{x}(t) \) is fed to a PLL model to produce the modulation signal \( i(t) \), that is then fed back into the physical system. The simulation components are described in more detail below, followed by performance and bit width evaluations.
A. Kalman Filter

1) System Model: The motion of the trapped nanoparticle can be modeled for each axis independently for small oscillation amplitudes [11]. Hence, we only describe the system model for the motion in the x-direction in the following. The motion in the x-direction is given by Newton’s second law as

\[ m \cdot \ddot{x}(t) = F_0(t) + F_d(t) + F_I(t) \]  

where \( m \) is the (constant) mass of the particle \( F_0(t) \) is the force due to the optical potential (cf. Fig. 3) which is to first-order harmonic, meaning that the potential energy of the particle is quadratic in its displacement from the origin. Accordingly, we model this force as

\[ F_0(t) = -k(t) \cdot x(t) = -m \cdot (\Omega_0 + \delta \Omega(t))^2 \cdot (1 + i(t)) \cdot x(t) \]

where \( k(t) \) is the trap stiffness and \( \Omega_0 \) denotes the eigenfrequency of the particle. The factor \((1+i(t)) \) in \( k(t) \) accounts for the trapping laser intensity modulation that varies the spring constant of the trapping potential. The function \( \delta \Omega(t) \) accounts for (slow) frequency drifts of the particle’s eigenfrequency \( \Omega_t \).

The term

\[ F_d(t) = -m \cdot \Gamma_0 \cdot \dot{x}(t) \]

in (1) accounts for damping by photons or air molecules, where \( \Gamma_0 \) denotes the damping rate. The term \( F_I(t) \) is a random fluctuation force [11], again arising from interactions of the particle with photons or air molecules, and is modeled as a Gaussian white noise process. The power of this Gaussian white noise is chosen such that in the absence of feedback, the center-of-mass temperature of the particle equilibrates with room temperature. The equation of motion (1) corresponds to a damped harmonic oscillator with the state space form

\[
\dot{\mathbf{x}}(t) = 
\begin{bmatrix}
0 \\
-(\Omega_0 + \delta \Omega(t))^2 \cdot (1 + i(t)) \\
-\Gamma_0
\end{bmatrix}
\begin{bmatrix}
x(t) \\
\dot{x}(t)
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
F_I(t) \\
0
\end{bmatrix}
\begin{bmatrix}
x(t) \\
\dot{x}(t)
\end{bmatrix}
\]

(2)

where \( \mathbf{x}(t) = [x(t), \dot{x}(t)]^T \). The measured position of the nanoparticle is described by \( \tilde{x}(t) \) in (2), where \( \mathbf{H} \) is the measurement matrix and \( \mathbf{w}(t) \) denotes the measurement noise that is modeled as a Gaussian white noise process with the same noise power as the photodetector.

2) Discrete Time Kalman Filter: In this paper, we employ a standard linear Kalman filter to estimate the state of the system model (2). However, due to the terms \( i(t) \) and \( \delta \Omega(t) \), the state transition matrix \( \mathbf{A}(t) \) is time dependent and has to be calculated in each time step. Acquisition of the modulation signal \( i(t) \) is straightforward in-sysm, but the real-time estimation of the frequency drift \( \delta \Omega(t) \) would add considerable design complexity. Therefore, the frequency drift \( \delta \Omega(t) \) is set to zero in the current implementation of the Kalman filter system. In Section V-E, we discuss the consequences of this choice and observe that this simplification is valid over the time horizon of short measurements on the order of seconds.

The discrete time system equations are given by

\[
\mathbf{x}_{k+1} = \mathbf{A}(k \cdot T_s) \mathbf{x}_k + \mathbf{v}_k \\
\mathbf{z}_k = \mathbf{x}_k + \mathbf{w}_k
\]

where \( \mathbf{v}_k \) and \( \mathbf{w}_k \) are the process and measurement noise signals with covariance \( \mathbf{Q}_k = \mathbb{E}[\mathbf{v}_k \mathbf{v}^T_k] \) and variance \( \mathbf{R} = \mathbb{E}[\mathbf{w}_k \mathbf{w}^T_k] \), respectively. The discrete state transition matrix \( \Phi_k \) is calculated in each time step according to

\[
\Phi_k = \exp(\mathbf{A}(k \cdot T_s) \cdot T_s) = \begin{bmatrix}
\Gamma_0 \sin(\omega_k T_s) + \cos(\omega_k T_s) & \omega_k^2 \sin(\omega_k T_s) \\
\frac{\omega_k}{2} \cos(\omega_k T_s) + \omega_k^2 \sin(\omega_k T_s) & 2 \omega_k \sin(\omega_k T_s)
\end{bmatrix}
\]

(3)

where \( T_s = 1/f_s \) is the sampling period and

\[
\omega_k = \sqrt{((\Omega_0^2(1 + i(k \cdot T_s))) - \Gamma_0^2)/4}
\]

is the discrete-time oscillation frequency of the system (under-damped case), which includes effects due to damping and modulation. The Kalman filter algorithm proceeds as follows [23], [24]: first, the next state is estimated according to

\[
\mathbf{x}_{k|k-1} = \mathbf{Φ}_{k-1} \mathbf{x}_{k-1|k-1}
\]

(4)

and the a priori error covariance matrix follows as

\[
\mathbf{P}_{k|k-1} = \mathbf{Φ}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{Φ}^T_{k-1} + \mathbf{Q}_k
\]

Then, the optimal Kalman gain is calculated

\[
\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R})^{-1}
\]

in order to update the state estimate according to

\[
\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \mathbf{x}_{k|k-1})
\]

where \( \mathbf{x}_k \) is the measured position signal. Note that the Kalman filter gain \( \mathbf{K}_k \) is scalar in this case, since the measured \( \mathbf{x}_k \) signal is scalar.

The final error covariance matrix is then given by

\[
\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}^T) + \mathbf{K}_k \mathbf{R} \mathbf{K}_k.
\]

This concludes one time step and the Kalman filter algorithm starts again with (4). The Kalman filter is initialized with \( \mathbf{x}_{0|0} = \mathbb{E}[\mathbf{x}_0] \) and \( \mathbf{P}_{0|0} = \mathbb{E}[(\mathbf{x}_0 - \mathbf{x}_{0|0})(\mathbf{x}_0 - \mathbf{x}_{0|0})^T] \).
B. Simulation Details

1) Physical System: The simulation flow diagram of the physical system is shown in Fig. 6. The random force \( f(t) \) is simulated with the Process Noise block and measurement noise is added to \( x(t) \), accounting for all experimentally relevant noise sources. The noise generators are band-limited White noise Simulink blocks.

2) ADCs and DACs: The analog-to-digital converter (ADC) blocks in Fig. 5 are realized with an appropriate analog anti-aliasing low-pass filter, a rate-transition block and a quantization block. The rate-transition block samples the continuous input signal with the system sampling frequency \( f_s \), and the ADC input is then quantized to \( b_{ADC} \) bit. The DAC block in Fig. 5 contains a rate-transition block and a quantization block that limits the resolution of the DAC converter to \( b_{DAC} \) bit.

3) Estimator: The Estimator block in Fig. 5 realizes two Kalman filter variants. The first variant is a standard Simulink block that is used as a reference. The second variant is our high-level synthesis (HLS) C++ implementation of the Kalman filter that will be synthesized for the target FPGA later (Section IV-C). We employ a MATLAB MEX S-Function to wrap the HLS Simulink blocks.

4) Phase-Locked Loop: The PLL block internals are depicted in Fig. 7. The PLL consists of three main components: the phase detector, the loop filter, and the voltage-controlled oscillator (VCO). The phase detector compares the input signal with the VCO’s output signal and generates a phase error signal. The loop filter removes unwanted frequency components of the phase error signal and can be used to adjust the overall PLL performance. An additional PID controller allows us to set the phase shift \( \theta \) to any desired value (phase setpoint). The frequency divider in the feedback path of this PLL model ensures that the output frequency is twice the input frequency, which is required by the parametric feedback scheme employed [2], [3]. Note that the frequency doubling is performed using a separate modulator after the PLL in the HF2LI device used in the experiments to derive the feedback signal [25]. Other than that, all components have been parameterized to reflect the actual configuration of the HF2LI device as accurately as possible.

C. Kalman Filter versus PLL-Only Approach—Simulation Results

To verify the performance of the Kalman filter approach, we compare two simulation runs (with/without Kalman filter), employing parameters that are representative of our real experimental setup (Table I). For this simulation, we use the maximum oscillation frequency of \( \Omega_0 = 2\pi \times 150 \text{ kHz} \) encountered in practice, since this represents the most challenging operating point for the HW system. The system parameters \( f_s \), \( b_{ADC} \), and \( b_{DAC} \) have been deliberately set to high values to minimize the implementation loss for this assessment (the impact of these parameters is evaluated later). For each simulation run, we tune the PLL and determine the optimal phase setpoint \( \theta \). Following [3], the cooling performance is then quantified using the center-of-mass temperature \( T_{c.m.} \).

\[
T_{c.m.} = \frac{1}{k_B} m \langle \Omega_0 + \Delta \Omega \rangle^2 \langle x^2 \rangle
\]

where \( \langle x^2 \rangle \) is the variance of the particle position and \( k_B \) is the Boltzmann’s constant. The oscillation frequency offset \( \Delta \Omega \) is small compared to the particle’s oscillation frequency \( \Omega_0 \).

As shown in Table II, our simulation predicts that the usage of a Kalman filter can reduce \( T_{c.m.} \) from 331 to 258 \( \mu \text{K} \), which is an improvement of about 20%. Fig. 8 further illustrates the time and frequency behavior of the simulation run with Kalman filter enabled. Fig. 8(a) shows 70 ms traces of the measurement signal \( x(t) \), the true position \( x(t) \) and the position estimated by the Kalman filter \( \hat{x}(t) \). In this simulation, the feedback loop is closed after 3 ms. Before this point, the position signal is oscillating with the initial amplitude of approximately 50 nm, corresponding to room temperature. As soon as the loop is closed, the oscillation amplitude decreases exponentially with a time constant of approximately 5 ms. Fig. 8(b) shows the power spectral density (PSD) of the three signals computed in the time window from 60 to 70 ms.
The measurement signal is at 1 pm.

The eigenfrequency of the system comprised of three main subsystems: the analog and digital subsystems on the FPGA board, and the laboratory equipment such as the PLL and modulators.

### TABLE II

<table>
<thead>
<tr>
<th>Metrics</th>
<th>$\sqrt{\langle \varepsilon \rangle}$</th>
<th>$T_{\text{rms}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman Filter OFF</td>
<td>45.1 pm</td>
<td>331 $\mu$K</td>
</tr>
<tr>
<td>Kalman Filter ON</td>
<td>39.8 pm</td>
<td>258 $\mu$K</td>
</tr>
</tbody>
</table>

![Fig. 8](image_url)  

Fig. 8. (a) Time traces and (b) PSDs from the Kalman filter simulation.

The PSDs of $x(t)$ and $\hat{x}(t)$ have a peak at the particle’s eigenfrequency $\Omega_0$ of $2\pi \times 150$ kHz, and the noise floor in the measurement signal is at 1 pm$/\text{Hz}$. The PSD peak is just slightly above the noise floor, which represents the ultimate performance limit (further improvements can only be achieved by reducing the noise floor, e.g., by increasing the photon detection efficiency).

#### D. Sampling Rate and Bit Width Evaluations

To properly parameterize the system, we run simulations with different values for $f_s$, $b_{\text{ADC}}$, and $b_{\text{DAC}}$ and assess the resulting cooling performance. The goal is to find parameterizations with negligible implementation loss with respect to double precision. In this simulation, the particle oscillation frequency is $\Omega_0 = 2\pi \times 150$ kHz and indeed, we can observe in Fig. 9(a) that at least a sampling rate of $2\Omega_0 / (2\pi) = 300 \text{kSample/s}$ is required such that the system performs correctly (steady-state motional standard deviation reduces from $\sim 1 \times 10^5$ to $\sim 40$ pm at a sampling rate of 300 kSample/s). Furthermore, the modulation signal $i(t)$ oscillates with twice that frequency (this signal is used to update the Kalman state transition matrix as described in Section IV-C), which mandates a sampling frequency of at least $f_s \geq 4\Omega_0 / (2\pi)$ to ensure proper conversion.

We can observe in Fig. 9(b) that the cooling performance degrades below an ADC resolution of $b_{\text{ADC}} = 9$ bit, where the quantization noise starts to dominate. Above 9 bit, the dominant noise source is the measurement noise from the photodetector. Likewise, we find that the DAC resolution should be at least $b_{\text{DAC}} = 16$ bit. Note that the HLS C++ Kalman filter implementation employs single-precision FP arithmetic since the algorithm requires many accurate matrix operations and computations of trigonometric functions. We found that the implementation loss with respect to the double-precision model is insignificant when using single-precision FP arithmetic.

Fig. 10 shows the block diagram of the developed cooling system comprised of three main subsystems: the analog subsystem including ADC and DAC implemented on an analog printed circuit board (PCB) designed by us, the digital subsystem mainly containing the Kalman filter implemented on an FPGA, and the other settings to sense and control the physical system. Fig. 11 shows the complete architecture of the designed system with Kalman-enhanced cooling in one of the three axes. The initial experimental setup shown in Fig. 1 is extended by an ADC acquisition subsystem bringing measured signals into the digital domain, an FPGA hosting the Kalman filter, a DAC subsystem, and a fourth PLL. The proposed system supports the original feedback scheme as well as the Kalman-enhanced feedback by adjusting the two gains $G_X$ and $G_K$, accordingly. Both the ADC and DAC subsystems are realized on a custom PCB that is connected to an FPGA development board featuring a Xilinx Virtex-7 FPGA via a 400 pin FPGA Mezzanine Card (FMC) connector. Communication between the FPGA and the converters is realized with serial peripheral interface (SPI). The AXI-Lite interface of the Kalman filter is attached to a Microblaze processor to enable configuration at runtime [via the universal asynchronous receiver–transmitter (UART) interface].

The system components are running at different operating frequencies (highlighted in Fig. 11), but the sampling frequency $f_s$ remains the same for all subsystems. All components work in a sample-pipelined manner, i.e., after each sampling period, the ADC interface hands on the next sample to the Kalman filter that in turn hands on the current state estimate to the DAC. Additional latency due to this pipelined processing is compensated in the PLL by setting the phase $\theta$ appropriately. The system is currently running with $f_s = 500 \text{kSample/s}$ limited by the achieved Kalman filter throughput. As evaluated in Section III, the sampling
frequency \( f_s \) needs to be at least four times larger than the oscillation frequency \( \Omega_0 \). Hence, the system currently supports experimental arrangements with oscillation frequencies up to \( \Omega_0 = 2\pi \times 500 \text{kHz}/4 = 2\pi \times 125 \text{kHz} \) which is sufficient for most in-field experiments.

**IV. SYSTEM ARCHITECTURE**

A. Photodetectors

The sensors employed in the experiments are so-called balanced photodetectors, which amplify the difference photocurrent between the two photodiodes. By means of a D-shaped mirror, the output light field is split and sent to both ports of the balanced photodetector, which leads to a signal proportional to the particle’s position along the measured axis. Balanced detection has the advantage that common mode noise associated with relative intensity noise of the laser is removed [2]. The employed detector is home-made and based on the photodiode ETX1000 from JDSU. The detector has a detection bandwidth of 350 kHz and provides a photon shot noise limited signal output for optical input powers between 1 and 100 mW. For the measurements shown in Fig. 9, we shine an overall optical power of 40 mW on the photodetector ETX1000 from JDSU. The detector has a removed [2]. The employed detector is home-made and based on the photodiode ETX1000 from JDSU. The detector has a detection bandwidth of 350 kHz and provides a photon shot noise limited signal output for optical input powers between 1 and 100 mW. For the measurements shown in Fig. 9, we shine an overall optical power of 40 mW on the balanced detector which translates to a spectrally flat measurement noise PSD \( S_{\text{meas}} \) of about 5 pm²/Hz.

B. Designed Analog Blocks

The analog subsystem comprises the ADC and DAC parts in Fig. 11 and is implemented on a separate PCB that is attached to the Virtex-7 FPGA board using a 400-pin FMC connector [26] using two SPI ports, one for the ADC and one for the DAC. Section IV-B.1 will give more details of the analog paths.

1) Analog-to-Digital Converter: The ADC subsystem consists of two chains, one for the photodetector output \( \tilde{x}(t) \) and one for the PLL output \( \tilde{i}(t) \). With the current sampling rate of \( f_s = 500 \text{ kSample/s} \), the two ADCs can resolve signals up to 250 kHz. The anti-aliasing filters are fourth-order Butterworth filters with optimized filter coefficients in order to ensure flat passband characteristics (cutoff has been set to 200 kHz). They are realized as two cascaded second-order Sallen–Key filters using low-noise operational amplifiers ADA4807-2 from Analog Devices. We chose to use an 18-bit ADC, AD4003 from Analog Devices, that provides plenty of headroom over the minimal required bit width of \( b_{\text{ADC}} = 9 \) bit. As this converter expects a differential input signal, the single-ended filter output signals are converted to differential signals via the differential amplifier, ADA4940-1 from Analog Devices.

2) DAC Subsystem: A target sampling rate of 500 kSample/s demands for DAC settling times well below 2 ns. This requirement limits the number of eligible DAC devices—especially when high resolution is crucial as indicated by the simulation results in Section III-D. Due to this limitation, a 16-bit DAC, the AD5541 from Analog Devices, has been chosen, meeting the requirements of \( b_{\text{DAC}} = 16 \) bit as well as featuring a settling time of 1 ns. Subsequent buffering of the DAC output is realized using an operational amplifier (AD820BRZ) followed by a reconstruction filter. Bandwidth requirements of the output signal are similar to the input channels, enabling us to reuse the same filter architecture and coefficients for the reconstruction filter. Finally, the output signal is connected to an SMA jack with a 50-Ω impedance enabling direct connection to the PLL.

C. FPGA Kalman Filter HW Implementation

The Kalman filter estimates the state of the particle in terms of position \( \hat{x}_k \) and velocity \( \dot{\hat{x}}_k \) based on the measured position \( \tilde{x}_k \). First, the innovation signal \( \tilde{x}_k = \hat{x}_k - \tilde{x}_k \), defined as the difference of the measured and the predicted position, is evaluated and scaled with the Kalman gain \( K_k \). Subsequently, the predicted position is updated with the resulting correction and, after a system delay, multiplied with the estimated state transition matrix \( \Phi_{k-1} \) [cf. (3)], resulting in a new prediction of the particles state. While the intensity modulation \( i(t) \) is measured in the proposed system and used to calculate \( \Phi_k \).
the other parameters including the nanoparticle’s oscillation frequency \( \Omega_0 \) and damping rate \( \Gamma_0 \) need to be configured separately. To facilitate this configuration process, we designed a MATLAB interface. The interface communicates with the FPGA via UART as mentioned in Section IV-A. It can initialize the Kalman filter with the parameters introduced earlier, and make it easy to read and rewrite these parameters, and reset or bypass the Kalman filter during the experiment.

For implementation, we adopt a Vivado HLS design flow where a C++ description of the algorithm is first converted to register-transfer level code, that is then synthesized and mapped onto the target FPGA. Such a design approach does not only enable system simulation of a bit-true high-level model of the later Kalman filter implementation but also enables fast prototyping and easy adaption to late specification changes. Due to the accuracy requirements, single-precision FP arithmetic has been used to implement the Kalman filter. The minimum latency of the Kalman filter is currently 196-clock cycles at 100 MHz, which enables system sampling rates up to 500 kSample/s.

The complete design including the Microblaze processor and interfaces requires around 49–k lookup tables (LUTs), 252 DSP slices and \( \sim 11 \) Mbit of block RAM, which corresponds to 16%, 9%, and 29% of the respective available resources, on the target Virtex-7 FPGA from Xilinx (XC7VX485T). There is, hence, enough space left for an extension to three axes, as the Kalman filter mainly uses LUTs and DSP slices. The block RAM is only used in the Microblaze processor. Fig. 12 shows a photograph of the HW system in field.

The system is hierarchically evaluated by starting from the analog subsystem in order to verify the basic functionality and make sure that the interfaces between the different parts are working. First, the ADC and DAC subsystems are separately verified by applying different voltages and comparing the digital values and vice versa. Second, we connected the analog subsystems to the FPGA, and bypassed the Kalman filter in the digital domain in order to make sure that the ADC to FPGA and FPGA to DAC interfaces are working properly. This test involved performing a voltage amplitude sweep and a frequency sweep. The results of these tests are plotted in Figs. 13 and 14, respectively. We can observe that these basic infrastructure subsystems work properly for the input voltages up to around 500 mV and for frequencies between 10 and 200 kHz. This is in line with the requirements, as the signal amplitudes and frequencies will be around 300 mV and 150 kHz, respectively, in the experiments. The frequency response shown in Fig. 14 matches the behavior of the filters including the low-pass antialiasing filter, reconstruction filter, and a high-pass digital filter implemented on FPGA which is used to remove the dc offset. In a final test, we performed several cooling tests in-field with the bypassed Kalman filter, with the parameters shown in Table III. Compared to the experimental setup without the FPGA system in the loop, no negative impact on cooling performance was noted.

### B. Basic Functional Verification

In this section, we verify the basic functionality of the system starting with the analog subsystem, show initial comparisons between a system with and without Kalman filter in the loop, and investigate the impact of the Kalman filter parametrization on cooling performance.

#### A. Experimental Setup

To initialize the system for evaluation of feedback performance, the vacuum chamber is evacuated after loading of a particle into the trap, followed by a calibration phase to obtain the parameters for the use of the Kalman filter as introduced in Section III-B1, including the damping rate \( \Gamma_0 \), particle’s oscillation frequency \( \Omega_0 \), the mass of nanoparticle \( m \), and the ratio for the conversion from the particle’s position to the value used in the digital domain. These parameters are slightly different for each experimental run and will be listed separately in the following.

#### C. Kalman Filter Verification Out of Loop

To verify the performance of the Kalman filter in an in-field experiment, we modify our experimental setup as illustrated in Fig. 15(a). We introduce a white noise source to artificially obscure the measurement record to then retrieve the
original position using the Kalman filter. Our detector records the position time-trace \( x(t) \) which is inevitably obscured by photon shot noise. In order to assess the performance of the Kalman filter, we add additional white Gaussian noise \( n \) to the measurement record to obtain the noisy signal \( \tilde{x}(t) = x(t) + n(t) \). The noisy signal \( \tilde{x}(t) \) is processed by the Kalman filter to yield the filtered time trace \( \hat{x}(t) \). Note that in this procedure, the added noise \( n \) is a factor of 20 larger than the inevitable photon shot noise on \( x \), such that in this section, we regard the measurement \( x \) as the idealized “true” position which the Kalman filter tries to recover from the noise-obscured record \( \tilde{x} \). During our experiments, the Kalman filter is out of the loop and the particle is cooled by the PLL only, which is fed by the artificially obscured signal \( \tilde{x} \).

In Fig. 15(b), we show the PSD of the noisy signal \( \tilde{x} \) as the red dashed line. The power spectrum shows a Lorentzian peak, arising from the particle motion, riding on top of a white noise floor arising from the added noise. For comparison, we plot the PSD of the Kalman filtered signal \( \hat{x} \) as the red solid line, which only consists of a Lorentzian, without a white background, indicating that the Kalman filter has indeed filtered out the noise. For further illustration, we show two additional PSDs for \( x \) (blue dashed line in Fig. 15) and \( \tilde{x} \) (blue solid line) for the case where no additional noise was added to the measurement record. Also, in this case, the Kalman filter reduces the out-of-band noise, which here is the photon shot-noise.

Importantly, a simple comparison of the PSDs of unfiltered and filtered signals does not yet demonstrate that the filtered signal indeed provides a faithful representation of the original measurement record. In order to assess the information content in the Kalman filtered signal \( \hat{x} \) relative to the original signal \( x \), we analyze the coherence of \( \hat{x} \) and \( x \) in Fig. 15(c) and plot it as the solid line. The coherence of two signals \( u \) and \( v \) is defined as

\[
C_{uv}(\omega) = \frac{|S_{uv}(\omega)|^2}{S_{uu}(\omega)S_{vv}(\omega)}
\]

(6)

where \( S_{uv}(\omega) \) is the cross-spectral density of the signals \( u \) and \( v \), estimated with the Welch method using Hanning windowing [27]. The coherence is a real-valued quantity which vanishes for perfectly uncorrelated signals and which assumes a value of one for perfectly correlated signals. We consider the coherence \( C_{xx}(\omega) \) [solid lines in Fig. 15(c)] as well as the coherence \( C_{\tilde{x}x}(\omega) \) [dashed lines in Fig. 15(c)]. Let us first consider the case where the noise added to the
neither the signal largely dominates over the noise. Off resonance, Fig. 15(c). Clearly, the coherences peak on resonance, where the measurement record is 20× the shot-noise level [red lines in Fig. 15(c)]. The pressure is 2 × 10⁻⁸ mbar. Starting from room temperature, the PLL reduces the particle’s center-of-mass temperature by five orders of magnitude to 1.2(1) mK. In case of the Kalman filter, 1.6(1) mK are reached. The respective temperatures are proportional to the areas under the curves. Every spectrum is based on a 5-s-long time trace.

Accordingly, the Kalman filter has suppressed the out-of-band noise (visible as the strongly reduced content of the PSD in Fig. 15(c)). In that case, one may be surprised by the falloff of coherence by the Kalman filter when analyzing the case of the eigenfrequency of the particle, which is around 147.1 kHz recalibrated accordingly. Furthermore, the system has been tested in a different pressure regime of ultralow vacuum. The nanoparticle’s oscillation frequency is about 146.5 kHz. The measurement noise floor of the photodetector corresponds to \( \sim 5 \times 10^{-25} \) m²/Hz in this setting. These parameters were then used to initialize the Kalman filter. The experiments are conducted under two different circumstances:

1) Conventional cooling with the PLL according to Fig. 1.
2) Kalman filter assisted cooling as shown in Fig. 11.

In both cases, we restrict our evaluation of the cooling performance to the x-axis. The y- and z-axes are continuously cooled using conventional PLL cooling. For each case, six 5 s long time traces are recorded and the corresponding power spectra \( S_{xy}(f) \) are shown in Fig. 16. The center-of-mass temperature \( T_{c.m.} \) can be estimated by integrating the spectra \( T_{c.m.} = m \Omega_0^2 \int d f S_{xy}(f) \) [28]. We find a cooling performance of about five orders of magnitude from 300 K (room temperature) to 1.2(1) mK by PLL cooling, as compared to 1.6(1) mK using Kalman filter assisted cooling. Accordingly, on the one hand, our measurements show that the Kalman filter system is in principle working. On the other hand, we conclude that the experimental implementation currently falls short of providing the performance increase by 20% predicted by our numerical modeling. In the following, we evaluate sources of error that currently limit the cooling performance of the Kalman filter assisted system.

### E. Parameter Tuning and Frequency Drifts

One of the practical complications and potential sources of error we experienced during our experiments is the tuning of the Kalman filter parameters. For instance, with a mismatch between the particle’s oscillation frequency (given by the trap stiffness) and the frequency parameter set for the Kalman filter, the nanoparticle can even be heated by the feedback signal instead of being cooled. This effect can be appreciated by reconsidering the schematic description of the feedback scheme in Fig. 3. Clearly, by phase shifting the modulation of the trap stiffness \( k(t) \), it is possible to transition from parametric cooling to parametric heating. We investigate the dependence of the cooling performance on the mismatch between the oscillation frequency \( \Omega_0 \) and the frequency parameter set for the Kalman filter in Fig. 17. This measurement is carried out at a pressure of \( \approx 10^{-5} \) mbar. We plot the variance of the particle position \( \langle x^2 \rangle \) for different settings of the frequency parameter of the Kalman filter. Lower variance means better cooling performance due to the relation \( T_{c.m.} = m \Omega_0^2 \langle x^2 \rangle / k_B \). The cooling performance is maximized when the frequency parameter of the Kalman filter is set to the eigenfrequency of the particle, which is around 147.1 kHz in the presented measurement. For comparison, we show the variance of the particle motion in the absence of feedback cooling as the red line in Fig. 17. We observe that when the frequency parameter of the Kalman filter strongly deviates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vacuum chamber pressure</td>
<td>( 2 \times 10^{-8} ) mbar</td>
</tr>
<tr>
<td>Eigenfrequency ( \Omega_0 )</td>
<td>( \approx 2\pi \times 147 ) kHz</td>
</tr>
<tr>
<td>Damping rate ( \Gamma_0 )</td>
<td>( \approx 2\pi \times 10^{-4} ) Hz</td>
</tr>
<tr>
<td>Particle mass</td>
<td>1.7 ( \times 10^{-19} ) kg</td>
</tr>
<tr>
<td>Measurement noise PSD ( S_{yy} )</td>
<td>( 5 \times 10^{-25} ) m²/Hz</td>
</tr>
</tbody>
</table>

**Table IV**

**In-Field Test Parameters**

![Fig. 16. Comparison of PLL and Kalman filter cooling performance. The spectra on the left are from a PLL cooled particle, the right ones are Kalman filter assisted. For each case, one spectrum is shown in red for better legibility. The pressure is 2 × 10⁻⁸ mbar. Starting from room temperature, the PLL reduces the particle’s center-of-mass temperature by five orders of magnitude to 1.2(1) mK. In case of the Kalman filter, 1.6(1) mK are reached. The respective temperatures are proportional to the areas under the curves. Every spectrum is based on a 5-s-long time trace.](image-url)
from the eigenfrequency of the particle, the particle motion is even heated instead of cooled by the feedback signal.

The measurement in Fig. 17 demonstrates that it is crucial to set the frequency parameter of the Kalman filter exactly to the oscillation frequency of the particle. Importantly, in practice, drifts in the experimental setup (in particular in the laser oscillation frequency of the particle) lead to slow drift $\delta \Omega(t)$ of the particle’s eigenfrequency. These drifts are on the order of 10 Hz/s, which has been observed in our initial tests in [13]. An example of the PSD of the output of the Kalman filter in a situation where the eigenfrequency has drifted away from the Kalman frequency parameter is shown in Fig. 18. Here, the frequency parameter for the Kalman filter has been set to 123.9 kHz, while the particle’s oscillation frequency is around 125 kHz, which eventually leads to significant loss of cooling performance. For short experiments, this issue is not detrimental, and the assumption made in Section III-A that frequency drifts $\delta \Omega(t) \approx 0$ approximately holds over the time scale of short measurements in the order of seconds. However, longer measurements that last several minutes suffer from effects of frequency drifts. This observation suggests that an additional frequency tracking mechanism able to dynamically estimate $\Omega(t) = \Omega_0 + \delta \Omega(t)$ and update the Kalman filter parameters could lead to better in-field performance.

VI. DISCUSSION AND CONCLUSION

In this paper, we present the evaluation of an FPGA-based system that incorporates a Kalman filter to control levitated nanoparticles along one motional axis. The designed system incorporates an accurate analog front-end with differential input acquiring data from photodetectors. After ADC conversion, a digital Kalman filter optimally estimates the nanoparticle state, which is then converted back to the analog domain and fed back to the physical system via a PLL for parametric cooling of the center-of-mass temperature of the nanoparticle.

The Kalman filter has been implemented on a Xilinx Virtex 7 FPGA using a C++-based HLS methodology. This design approach has the advantage of enabling bit-true full system simulations based on the high-level C++ description, and it enables fast prototyping and easy algorithmic adaption to late specification changes. Our MATLAB Simulink-based full-system simulation indicates that a 20% lower motional center-of-mass temperature can be achieved using a Kalman filter.

Experimentally, we verify the basic functionality of the FPGA system and show that it is working correctly. Comparisons between first in-field measurements of the highly optimized PLL approach and our Kalman-based system revealed that careful tuning of the Kalman filter parameters (like the particle oscillation frequency) at runtime is crucial for good performance. In practice, an adaptive version of the filter, where system parameters are monitored and continuously updated, could provide a remedy to this issue. On the other hand, with all parameters correctly set, we have demonstrated in Fig. 15 that the Kalman filter produces a faithful representation of the particle position signal. We therefore believe that it is necessary to also consider other parts of the system in order to identify possible sources of error accounting for the discrepancy between simulation results and measurements. We note that our model does not incorporate the technical noise possibly added by the high-voltage amplifier addressing the EOM. This electronic noise would lead to fluctuations of the optical trap stiffness which could lead to a parametric heating of the particle motion. In addition, the Simulink model of the PLL used in our simulations differs in some technical details from the PLL implementation in the Zürich Instruments PLL used in the experiments.

Finally, we would like to spell out several questions of more fundamental character, triggered by our results. Technical issues regarding the optimization of the involved parameters aside, we believe that it is a relevant question to determine whether in principle a Kalman filter is superior to a PLL in our application. After all, a PLL is afilter designed to lock to a harmonic signal. It is an open question in how far a Kalman filter designed to predict the motion of a harmonic oscillator differs in its estimated output from the output of a PLL that is directly locked to the noisy harmonic oscillator signal. Following this thought, one might question whether...
feeding a Kalman-filtered signal into a PLL is an optimal approach. Indeed, it might be beneficial to directly derive a feedback signal from the Kalman filtered record of the particle motion in a fashion not relying on a PLL.

ACKNOWLEDGMENT

The authors would like to thank Zürich Instruments for providing them with the PLL parameters of their HF2LI instrument.

REFERENCES


Jiawei Liao received the B.Eng. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, and the University of Birmingham, Birmingham, U.K., in 2016, the M.Sc. degree in electrical engineering and information technology from ETH Zürich, Zürich, Switzerland, in 2018, where he is currently pursuing the Ph.D. degree with the Integrated System Laboratory. His current research interests include digital integrated circuit design, digital signal processing, embedded systems, and machine learning.

Michael Schaffner received the B.Sc. degree from the University of Applied Sciences Rapperswil, Rapperswil-Jona, Switzerland, and the M.Sc. degree in electrical engineering and information technology from the Swiss Federal Institute of Technology, Zürich, Switzerland, in 2012 and 2017, respectively. He was a Research and Development Engineer with Sensirion, Stäfa, Switzerland, where he was involved in integrating temperature and humidity sensors into smartphones. He is currently an Embedded Software Engineer with Duagon, Dietikon, Switzerland.

Michael Schaffner received the M.Sc. and Ph.D. degrees from the Swiss Federal Institute of Technology, Zürich, Switzerland, in 2012 and 2017, respectively. From 2012 to 2017, he was a Research Assistant with the Integrated Systems Laboratory and Disney Research, Zürich, where he is currently a Post-Doctoral Researcher with the Integrated Systems Laboratory. His current research interests include digital signal processing, video processing, and the design of very large scale integration circuits and systems. Dr. Schaffner was a recipient of the ETH Medal for his Diploma thesis in 2013.

Michele Magnó (SM’13) received the master’s and Ph.D. degrees in electronic engineering from the University of Bologna, Bologna, Italy, in 2004 and 2010, respectively. He is currently a Post-Doctoral Researcher with ETH Zürich, Zürich, Switzerland, and a Research Fellow with the University of Birmingham. He has authored more than 100 papers in international journals and conferences. His current research interests include wireless sensor networks, wearable devices, energy harvesting, low-power management techniques, and extension of lifetime of batteries-operating devices.

Matthias Korb (M’09) received the master’s and Ph.D. degrees in electrical engineering from Rheinisch-Westfälische Technische Hochschule Aachen University, Aachen, Germany, in 2006 and 2012, respectively. From 2012 to 2016, he was with the Research Group DSP Microelectronics, Office of the CTO, Broadcom Corp., Irvine, CA, USA, where he was involved in delivering dedicated digital IP cores for various communication systems of Broadcom’s product portfolio and developing next-generation wireless communication standards such as IEEE 802.11ax. Since 2016, he has been a Senior Researcher with the Integrated Systems Laboratory, ETH Zürich, Zürich, Switzerland.
Luca Benini (F'07) is currently the Chair of digital circuits and systems with ETH Zürich, Zürich, Switzerland, and a Full Professor with the Universita di Bologna, Bologna, Italy. He has authored or co-authored more than 900 peer-reviewed papers and five books. His current research interests include energy-efficient computing systems design, from embedded to high performance, and the design of ultralow power VLSI circuits and smart sensing microsystems.

Mr. Benini is a Fellow of the ACM and a member of the Academia Europaea. He was a recipient of the 2016 IEEE CAS Mac Van Valkenburg Award.

Felix Tebbenjohanns received the M.S. degree in electrical engineering and information technology from ETH Zürich, Zürich, Switzerland, in 2017, where he is currently pursuing the Ph.D. degree with the Photonics Laboratory.

His current research interests include exploring the ultimate limits of feedback cooling the center-of-mass motion of a levitated nanoparticle with the goal of bringing it as close as possible to the quantum ground state.

Rene Reimann received the M.A. degree in physics from Stony Brook University, Stony Brook, NY, USA, with a focus on quantum gases in state-dependent optical potentials, and the Ph.D. degree with Bonn University, Bonn, Germany, with the focus on cooling and cooperative coupling of single atoms in an optical cavity.

Since 2015, he has been a Senior Scientist with the Photonics Laboratory, ETH Zürich, Zürich, Switzerland, where he supervises and conducts experiments on levitated optomechanics, especially controlling external and internal degrees of freedom of levitated nanoparticles.

Vijay Jain received the Ph.D. degree in electrical engineering from ETH Zürich, Zürich, Switzerland, in 2017, with a focus on precision measurement methods and control protocols for optically levitated nanoparticles.

He is a Post-Doctoral Associate with the Yale Quantum Institute, where he is involved in the study of quantum acoustics with superconducting circuits.

Michael Gross received the M.Sc. degree in electrical engineering from the Swiss Federal Institute of Technology in Zurich (ETH Zurich), Zurich, Switzerland, in 2017, with a focus on control theory, optics, and electronics. His master thesis was on hardware-near software implementation of optimal state estimation for a levitated nanoparticle.

He is currently pursuing a career in the field of risk modeling, data science, and artificial intelligence.

Andrei Militaru was born in Bucharest, Romania, in 1992. He received the B.Sc. degree in electronics engineering from the Polytechnic University of Milan, Milan, Italy, in 2015, and the M.Sc. degree in electrical engineering from ETH Zürich, Zürich, Switzerland, in 2017, with a focus on optics and particle trapping, where he is currently pursuing the Ph.D. degree with the Photonics Laboratory, with a focus on optically levitated nanoparticles, led by Prof. L. Novotny.

Martin Frimmer received the Ph.D. degree from the FOM Institute AMOLF, University of Amsterdam, Amsterdam, The Netherlands, with a focus on spontaneous emission control.

He is currently a Scientist with the Photonics Laboratory, ETH Zürich, Zürich, Switzerland. His current research interest include interaction of light and matter on the subwavelength scale.

Lukas Novotny received the Ph.D. degree from ETH Zürich, Zürich, Switzerland, in 1996. From 1996 to 1999, he was a Post-Doctoral Fellow with the Pacific Northwest National Laboratory, Richland, WA, USA, where he was involved in new schemes of single-molecule detection and nonlinear spectroscopy. In 1999, he joined the Faculty of the Institute of Optics, University of Rochester, Rochester, NY, USA, where he started one of the first research programs with focus on nanooptics. He is currently a Professor of photonics with ETH Zürich. He has authored the textbook Principles of Nano-Optics (2nd edition). His current research interests include understanding and controlling light-matter interactions on the nanometer scale.

Dr. Novotny is a Fellow of the Optical Society of America and the American Association for the Advancement of Science.