Review of Scientific Instruments

The Raspberry Pi auto-aligner: Machine learning for automated alignment of laser beams

Cite as: Rev. Sci. Instrum. **92**, 015117 (2021); https://doi.org/10.1063/5.0032588
Submitted: 12 October 2020 . Accepted: 22 December 2020 . Published Online: 19 January 2021

Renju S. Mathew, Roshan O'Donnell, Danielle Pizzey, and Ifan G. Hughes

Review of Scientific Instruments | ARTICLE | scitation.org/journal/rsi

The Raspberry Pi auto-aligner: Machine learning for automated alignment of laser beams

Cite as: Rev. Sci. Instrum. 92, 015117 (2021); doi: 10.1063/5.0032588
Submitted: 12 October 2020 • Accepted: 22 December 2020 •
Published Online: 19 January 2021

Renju S. Mathew, Roshan O'Donnell, Danielle Pizzey,[a] and Ifan G. Hughes

AFFILIATIONS

Joint Quantum Centre (JQC) Durham-Newcastle, Department of Physics, Durham University, South Road, Durham DH1 3LE, United Kingdom

[a]Author to whom correspondence should be addressed: danielle.boddy@durham.ac.uk

ABSTRACT

We present a novel solution to automated beam alignment optimization. This device is based on a Raspberry Pi computer, stepper motors, commercial optomechanics and electronic devices, and the open-source machine learning algorithm M-LOOP. We provide schematic drawings for the custom hardware necessary to operate the device and discuss diagnostic techniques to determine the performance. The beam auto-aligning device has been used to improve the alignment of a laser beam into a single-mode optical fiber from manually optimized fiber alignment, with an iteration time of typically 20 minutes. We present example data of one such measurement to illustrate device performance.

I. INTRODUCTION

Machine learning (ML) methods can discover patterns in data without requiring any assumptions about the data's structure.[1] Performing research with ML began in earnest in the 1980s,[2] and by 1992, ML methods were used to, for example, create non-intuitive laser pulse-sequences for exciting rotational quantum states.[3] However, it is only in the last decade or so that ML methods have begun to be used more widely in the atomic, molecular, and optical (AMO) physics community. ML techniques have been used to create self-tuning, mode-locked lasers;[4–6] for automating the production of Bose–Einstein condensation;[7] and maintaining doughnut-shaped beams in scattering media.[8] ML has recently even been used to create new quantum experiments: the system both learned to create a variety of entangled states and improved the efficiency of their realization.[9] Despite these advances, no work using ML for beam alignment has been found.

"Walking the beam" is the process of aligning a laser beam using two adjustable mirrors in such a way that it will reach a specific point in space with a specific angle. It can be a laborious task to properly adjust the four knobs that control the horizontal (yaw) and vertical (pitch) angles of the two mirrors. Without making the correct sequence of adjustments, it is even possible to move further away from the goal instead of moving toward it. Aligning the signal into the detector is more difficult in experiments with very low light levels that necessitate using fiber-coupled photon-detectors,[10–13] as is often the case in quantum optics experiments, e.g., single-photon generation from hot thermal vapors.[14–24] Because of these issues, we were strongly motivated to commission a device to automate beam alignment. Not only can this device save time and find a more optimal alignment but alignment automation can also be a helpful addition to laser-safety protocols, an issue that is becoming increasingly important with the ever more widespread use of high-powered lasers.

In the same time frame that ML techniques have been utilized in laboratories, there has been a surge of interest in using low-cost but high-performance hardware, particularly in optics and imaging experiments.[25–31] Here, we present a device based on using stepper motors operated by a Raspberry Pi computer to control commercial kinematic mounts. By attaching a motor to the yaw and pitch knobs of each mount, the orientation of the mirror can be controlled electronically. We employ the machine learning algorithm M-LOOP[7] to optimize the coupling of a laser beam into a single-mode optical fiber, using a detector. We call this entire setup a "beam auto-aligner."

The main advantage of our design over the current commercial alternatives is the ability to modify the hardware and the ease of implementation. Since commercial hardware and software are proprietary, the equipment must be used as a black-box, limiting room for customization. Commercial software may need to be bought separately from the hardware and need periodic updating. For our beam auto-aligner, all of the hardware drawings, computer-aided design (CAD) files, and electronic schematics including a bill-of-materials are available in the supplementary material. The computer software, M-LOOP, to operate the device is available on the M-LOOP website[32] and is open-source.

The remainder of the paper is organized as follows: in Sec. II A, we present the hardware; Sec. II B discusses the relevant features of the M-LOOP software; we illustrate the performance of the beam auto-aligner in Sec. III; finally, conclusions are drawn and an outlook is provided in Sec. IV.

II. METHODS

A. Hardware

Figure 1 shows a rendering of the device. A NEMA-17 stepper motor attaches to each kinematic mount knob via a custom-made metal motor-mirror coupler, both of which are supported by a metal L-plate support mount. The L-plate support mount is not shown in the rendering drawing, but it is shown in a photograph inset.

The computer used to run M-LOOP and to control the motors is a Raspberry Pi 4 Model B, although any version of Pi will be suitable. The device has minimal footprint, with dimensions 85 × 56 × 17 mm^3, that can be placed directly on the optical table.

The output for M-LOOP to optimize is chosen to be the beam power on the output of a single-mode optical fiber. The optical beam power can be measured as a voltage on a photo-diode, a commercial power-meter, or an integrated photon counter. In this work, we use a homebuilt photo-diode circuit. An analog-to-digital converter (ADC) is required to convert the analog output of the photo-diode circuit into a form usable by the Pi; the 16-bit ADS1115 ADC is used in this work. The Raspberry Pi can operate four motors, which is adequate for this investigation. However, should the user require control of more than four motors, shift registers can be used in conjunction with the Raspberry Pi to increase the number of GPIO pins of the Pi. In addition, drivers are necessary as an intermediary between the Raspberry Pi, which output tens of milliamps, and the stepper motors, which require several amps. The L298N drivers are used in this work as they are able to switch the direction of current supplied to the motors and hence change the direction of rotation rapidly. The L298N drivers require 5 V–35 V, but to achieve maximum motor speed, a voltage of 30 V–35 V is advised. The Raspberry Pi requires 5 V power, and it powers the ADC and, if necessary, shift registers from this supply. The wiring for the electronic components is provided in the supplementary material.

Great care has been taken to design the hardware such that it does not suffer from hysteresis, specifically the motor-mirror coupler. Furthermore, the device should be used in conjunction with stable optomechanics: stable mirror mounts (SR100-100-2-BU) from Photonic Technologies (LiOp-Tec) and stable pedestals (RDS-MNI-P-75) and holding forks (RDS-MNI-HF-M) from Radiant Dye Laser are used in this work.[33] Should the reader wish to use a modified hardware design to suit their purpose, they should follow the troubleshooting advice provided in the supplementary material.

To confirm the device does not suffer from hysteresis, the reproducibility of the beam auto-aligner design for two axes (i.e., two knobs on one kinematic mirror mount) is shown in Fig. 2.

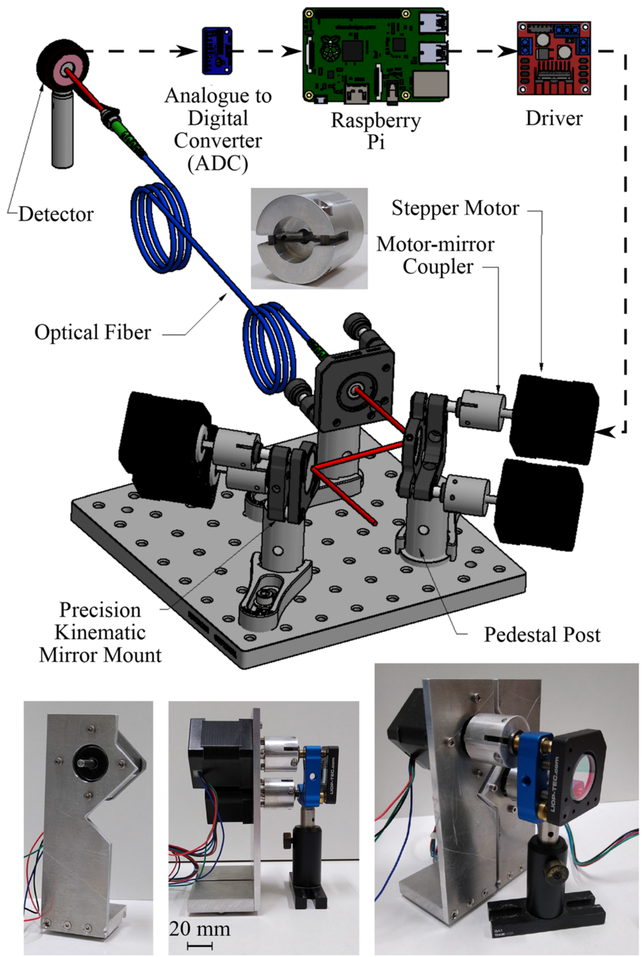

FIG. 1. Top: Schematic illustration of the beam auto-aligner assembly, showing a typical use scenario where four motors are used to control the pitch and yaw on two kinematic mirror mounts to align a laser beam into a single-mode optical fiber. The laser beam that exits the fiber is directed onto a detector, with the voltage reading inputted to the Raspberry Pi via an analog-to-digital converter (ADC). Should a photon counter be used as the detector, the ADC is no longer required. The dashed black arrows convey the electrical path. For scale, the breadboard the beam auto-aligner is mounted on is 115 × 115 mm^2. See the main text for details of the labeled components. Connecting cables and the support mount for the motor-mirror couplers are not shown in the rendering drawing for clarity. The inset of the rendering drawing is a photograph of the motor-mirror coupler. Bottom: Photographs of the hardware. Left: The L-plate motor support mount for one stepper motor. Middle and right: Illustrates how the motor attaches to the kinematic mirror mount knob via the motor-mirror coupler.

Starting from manually optimized fiber coupling, M-LOOP sends commands to the motors such that the mirror is moved back and forth between the starting orientation and some random orientation and the beam power coupled into the single-mode optical fiber is measured.

For the specific application of coupling light into a single-mode optical fiber, the transmitted power is expected to be a smooth, single-peaked function of mirror displacement.[34] The shape of the experimentally measured surface evident in Fig. 2 confirms

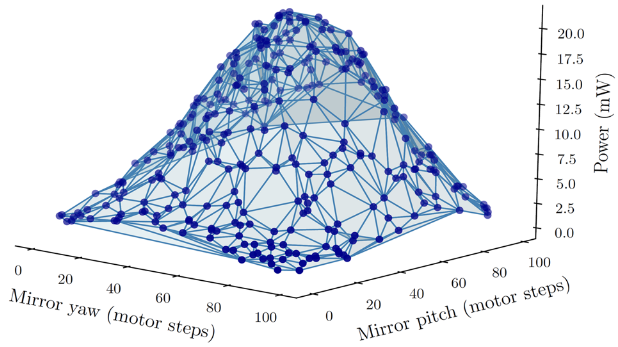

FIG. 2. Investigation of the reproducibility of the device for two axes on one kinematic mirror mount. The mirror yaw and pitch orientations are initialized by hand such that the coupling efficiency into the single-mode optical fiber is maximized. The motor moves the mirror to a random orientation, as shown by the blue circles, and immediately moves back to the starting position. This is repeated many times. A landscape is mapped out.

that the motor-mirror setup is working as required; the measured landscape would not have been observed in a system exhibiting hysteresis.

B. Software

The computer software, M-LOOP, to operate the device is available on the M-LOOP website.[32] The website provides basic instructions on how to use the software, including how to start M-LOOP through the terminal. M-LOOP is installed on the Raspberry Pi and is controlled using terminal commands on the Pi's operating system, although it is possible to use a custom Python interface.

M-LOOP contains several ML algorithms of which only Gaussian processes (GPs) were investigated in this work. The algorithm attempts to minimize a "cost," which is analogous to the sum of squared residuals in least-squares error analysis techniques.[35] Since we are interested in maximizing the power out of the fiber, P_{out}, the cost is set to equal the negative of the power $-P_{\mathrm{out}}$. M-LOOP is initialized with the boundary values of the parameter space it should explore. We have four parameters: a pitch (parameter 1, p_1) and yaw (parameter 2, p_2) for mirror 1 and a pitch (parameter 3, p_3) and yaw (parameter 4, p_4) for mirror 2. The boundary values for each parameter are defined such that one full 360° rotation of the kinematic mount knob is permitted—the values will need to be determined by the user when setting up the device as they will vary depending on the step size of the stepper motors and the thread pitch of the kinematic mounts used (see the supplementary material for more information). There is some unique set of values (p_1, p_2, p_3, p_4) where the coupling efficiency is at its maximum that M-LOOP will try to find. The "beam auto-aligner" completes its optimization cycle when one of the three halting conditions is met in M-LOOP: (1) The maximum number of runs has been reached. (2) The minimum cost value has been reached. (3) A certain number of runs has elapsed without a lower cost being found. In all instances, the user sets these values before the first run.

Figure 3 is a direct graphical output from the M-LOOP software when starting from poor fiber coupling. The user does not

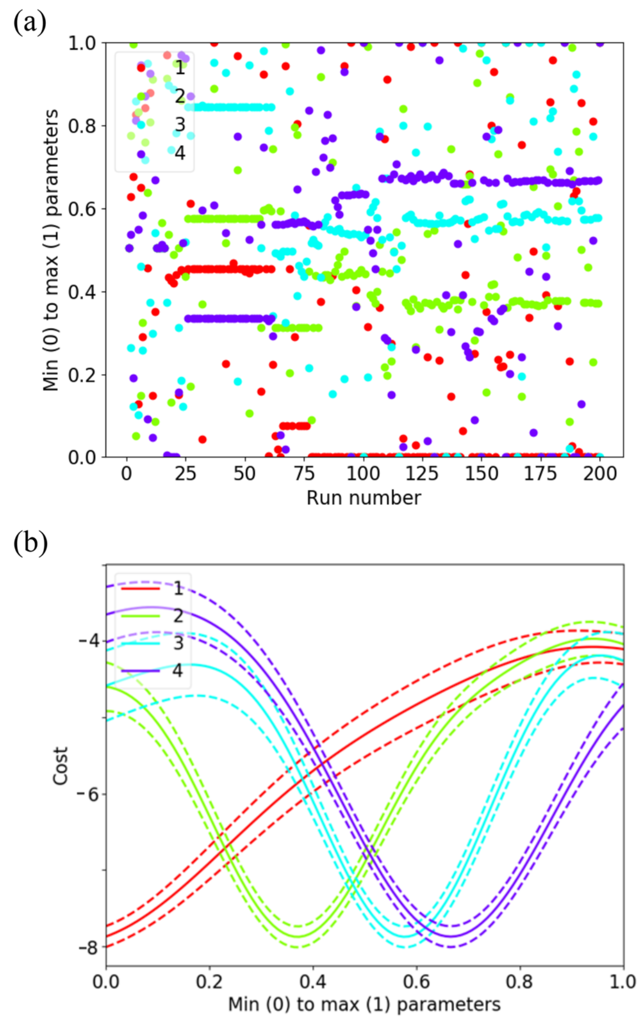

FIG. 3. Direct graphical output from the M-LOOP software when starting from poor fiber coupling. The parameter values labeled 1, 2, 3, and 4 in the legend are red, green, blue, and purple in the figures, respectively. The parameters correspond to the pitch ("1") and yaw ("2") for mirror 1 and the pitch ("3") and yaw ("4") for mirror 2. (a) The *y*-axis is the normalized number of turns on a kinematic mirror mount knob where 0.5 means no change from the original position. After a certain number of runs, M-LOOP homes in on the best parameters. For this dataset, the best value for parameter 1 (red) lay outside the range the user allowed M-LOOP to explore. (b) With a sufficient number of runs, M-LOOP builds an internal model of the parameter landscape. The dashed lines indicate error boundaries. Smooth curves indicate that the device is functioning as expected. For this dataset, the minimum of parameter 1 lay outside the range the user allowed M-LOOP to explore.

observe the plotting of this figure in real-time; it is only once M-LOOP has met one of the three halting conditions that the user obtains these results. Careful study of the output gives insight into what M-LOOP is doing at each run and gives indications on how to troubleshoot. For example, we can see that as shown in Fig. 3(a), for the first 25 runs, a random set of parameters is tried. This is typical, although the number of runs during this random search varies. By run 25, M-LOOP has homed in on an optimum set of parameter values as shown by the plateaus for all four parameters. It continues testing the nearby parameter space but regularly tries completely different parameters to ensure that it is not stuck in a local cost minimum.[35] Indeed, by run 70, it has discovered a different set of optimal parameters. However, by run 77, parameter 1 (red) is at the very edge of its constraint. The algorithm "wants" to explore values below 0, but our initial boundary conditions do not allow it. This suggests that we have set the boundary of the parameter space for parameter 1 too narrowly and that we should re-run the whole process with this boundary expanded. This is also evident in Fig. 3(b), which shows the predicted landscape of cost against the parameter value for the same experimental set shown in Fig. 3(a). The minimum of the parameter 1 curve (red) seems to lie just outside of the range that the user has allowed M-LOOP to investigate, as can be seen if we imagine extrapolating the red curve into the region of the negative x-axis. Nevertheless, the smooth curves for each of the four parameters demonstrate that our program is working as expected. M-LOOP takes ~20 minutes from start to finish for 200 runs.

III. RESULTS

In the remainder of the paper, we illustrate the use of the beam auto-aligner. For ease of direct comparison between initial starting conditions, the parameter values are normalized as shown in Fig. 4, with each sub-figure having its own normalization, so that the parameters only ever takes a value between −1 and 1.

Figure 4 illustrates what happens when we start the machine learning after manually aligning the fiber to what we, the user, believe to be the best coupling efficiency. As shown in Fig. 4(a), as expected, M-LOOP does a random exploration of the parameter space for 10 runs and quickly finds that the original location (0, 0, 0, 0) is optimal. M-LOOP nevertheless tests to see if there are any better values for these parameters until on run 55, it decides that indeed (0, 0, 0, 0) is the best case. Although M-LOOP continues to test further, it does not find better parameters. Figure 4(b) is the case where the auto-aligner works exactly as it was designed to do. Between runs 15 to 50, M-LOOP agrees with the user until it discovers an even better set of parameter values.

M-LOOP was able to get the same, if not slightly better, coupling efficiency than an experienced experimenter might achieve starting from poor fiber coupling. The purpose of this device is more for continual auto-optimization, rather than using the device to couple into a fiber from scratch. Once M-LOOP has found optimum parameters, the user can turn off the power to the motors, Raspberry Pi, and associated electronics; the mirror position and angle will remain as they were at the end of the M-LOOP optimization cycle. The user can then use the device for continual auto-optimization, e.g., the device could be set to perform auto-alignment at a set time every morning before an experiment is carried out. Alternatively, the device could remain powered, and the user could initiate remote alignment as the Pi can be controlled via the intranet using the commonly used Secure Shell (SSH) protocol, which is inbuilt into the Pi's Linux operating system.

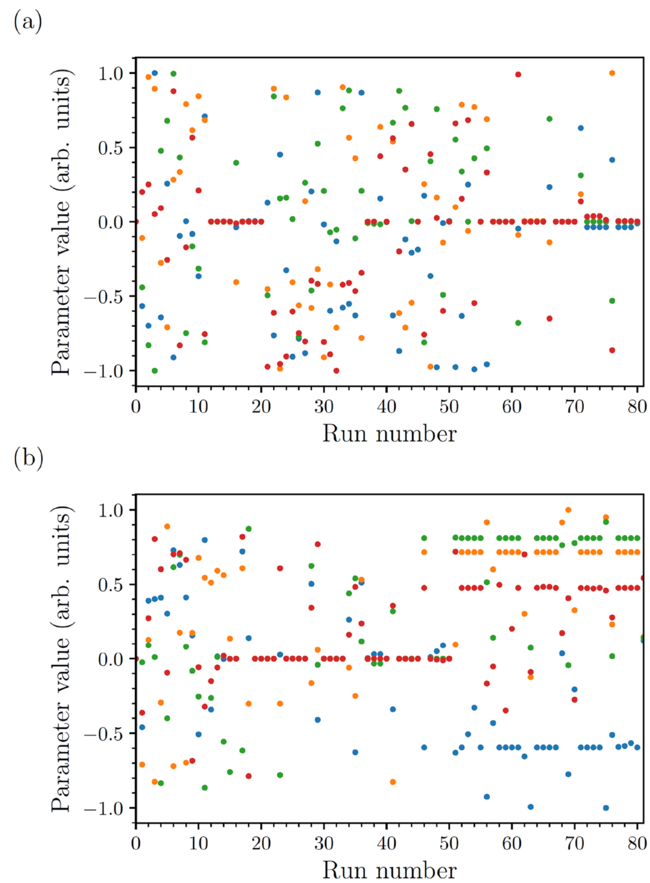

FIG. 4. Starting from the best (manually aligned) fiber coupling illustrating M-LOOP in action: (a) M-LOOP returns to the best coupling and (b) M-LOOP finds a better set of parameters (runs 50+). The parameter values are normalized, with each sub-figure having its own normalization. In all examples, parameters p_1, p_2, p_3, and p_4 are represented by blue, red, orange, and green, respectively.

The device is not limited to optimizing the laser power into an optical fiber; other applications include overlapping beams in a pump-probe experiment, where the peak-to-peak voltage of the signal would replace the photo-diode voltage of the beam power measured on the output of the optical fiber. There is the issue of scaling, however. The more parameters M-LOOP has to optimize, the longer it will take. Typically for our beam auto-aligner device, it took 20 minutes for M-LOOP to optimize four parameters (i.e., two mirrors having two control knobs each). If we were to use this device in a four-wave mixing experiment, where the signal depends on the overlap of three laser beams, M-LOOP would need to optimize 12 parameters (i.e., each laser beam having two mirrors for alignment control and each mirror having two control knobs). For Gaussian processes, the machine algorithm investigated in this work, the computational times scale cubically with the number of data;[36] this is not to say the device will not work to include more than four parameters but only that we have not investigated other machine learning algorithms to facilitate this.

IV. CONCLUSION

In conclusion, we have demonstrated how the laborious and time-consuming task of manually aligning beams into single-mode optical fibers can be automated using our beam auto-aligner, which implements the machine learning algorithm M-LOOP. The device is not limited to optimizing laser beam power into optical fibers; other applications include overlapping beams in a pump-probe experiment or aligning high-power dipole trap laser beams with an atomic cloud in cold-atom experiments, to name but a few. The intended use of the device is for continual auto-optimization, rather than for aligning laser beams from the initial setup. In addition, due to the functionality of the Pi, remote alignment can easily be performed.

SUPPLEMENTARY MATERIAL

In addition to M-LOOP's GitHub repository,[32] see the supplementary material for hardware CAD drawings, parts list (including a bill-of-materials), and electronic schematics.

ACKNOWLEDGMENTS

We gratefully acknowledge Nicholas Spong for providing support in getting the project off of the ground, Clare Higgins for testing the device and for editorial discussions, Thomas Cutler, Lina Marieth Hoyos-Campo, and Ana Rakonjac for editorial discussions, Aidan Hindmarch for sourcing equipment, the anonymous referees for constructive feedback, and the EPSRC (Grant No. EP/R002061/1) for funding.

DATA AVAILABILITY

The data that support the findings of this study are openly available at Durham Research Online (DRO): http://doi.org/10.15128/r1qr46r0844.

REFERENCES

[1] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Adaptive Computation and Machine Learning (MIT Press, 2012).
[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Adaptive Computation and Machine Learning (MIT Press, 2016).
[3] R. S. Judson and H. Rabitz, Phys. Rev. Lett. **68**, 1500 (1992).
[4] J. N. Kutz, X. Fu, and S. Brunton, *Advanced Photonics* (Optical Society of America, 2014), p. NTu4A.7.
[5] U. Andral, J. Buguet, R. Si Fodil, F. Amrani, F. Billard, E. Hertz, and P. Grelu, J. Opt. Soc. Am. B **33**, 825 (2016).
[6] T. Baumeister, S. L. Brunton, and J. Nathan Kutz, J. Opt. Soc. Am. B **35**, 617 (2018).
[7] P. B. Wigley, P. J. Everitt, A. van den Hengel, J. W. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. D. Quinlivan, P. Manju, C. C. N. Kuhn, I. R. Petersen, A. N. Luiten, J. J. Hope, N. P. Robins, and M. R. Hush, Sci. Rep. **6**, 25890 (2016).
[8] Y. Zhang, C. Wu, Y. Song, K. Si, Y. Zheng, L. Hu, J. Chen, L. Tang, and W. Gong, Opt. Express **27**, 16871 (2016).
[9] A. A. Melnikov, H. P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Proc. Natl. Acad. Sci. U. S. A. **27**, 16871 (2018).
[10] D. J. Whiting, N. Šibalić, J. Keaveney, C. S. Adams, and I. G. Hughes, Phys. Rev. Lett. **118**, 253601 (2017).
[11] J. Park, H. Kim, and H. S. Moon, Phys. Rev. Lett. **122**, 143601 (2019).
[12] C. Wang, C.-H. Lee, and Y.-H. Kim, Opt. Express **27**, 34611 (2019).
[13] J. Mika and L. Slodička, J. Phys. B: At., Mol. Opt. Phys. **53**, 145501 (2020).
[14] R. T. Willis, F. E. Becerra, L. A. Orozco, and S. L. Rolston, Phys. Rev. A **79**, 033814 (2009).
[15] R. T. Willis, F. E. Becerra, L. A. Orozco, and S. L. Rolston, Phys. Rev. A **82**, 053842 (2010).
[16] R. T. Willis, F. E. Becerra, L. A. Orozco, and S. L. Rolston, Opt. Express **19**, 14632 (2011).
[17] D.-S. Ding, Z.-Y. Zhou, B.-S. Shi, X.-B. Zou, and G.-C. Guo, Opt. Express **20**, 11433 (2012).
[18] A. MacRae, T. Brannan, R. Achal, and A. I. Lvovsky, Phys. Rev. Lett. **109**, 033601 (2012).
[19] D. J. Whiting, J. Keaveney, C. S. Adams, and I. G. Hughes, Phys. Rev. A **93**, 043854 (2016).
[20] Y.-S. Lee, S. M. Lee, H. Kim, and H. S. Moon, Opt. Express **24**, 28083 (2016).
[21] M. Parniak, A. Leszczyński, and W. Wasilewski, Phys. Rev. A **93**, 053821 (2016).
[22] D. J. Whiting, R. S. Mathew, J. Keaveney, C. S. Adams, and I. G. Hughes, J. Mod. Opt. **65**, 713 (2018).
[23] T. Jeong, J. Park, and H. S. Moon, Phys. Rev. A **100**, 033818 (2019).
[24] T. Jeong and H. S. Moon, Opt. Express **28**, 3985 (2020).
[25] N. B. Clayburn, K. W. Trantham, M. Dunn, and T. J. Gay, Rev. Sci. Instrum. **87**, 124903 (2016).
[26] J. P. Sharkey, D. C. W. Foo, A. Kabla, J. J. Baumberg, and R. W. Bowman, Rev. Sci. Instrum. **87**, 025104 (2016).
[27] D. J. Nemchick, B. J. Drouin, M. J. Cich, T. Crawford, A. J. Tang, Y. Kim, T. J. Reck, E. T. Schlecht, M.-C. F. Chang, and G. Virbila, Rev. Sci. Instrum. **89**, 073109 (2018).
[28] J. Keaveney, Rev. Sci. Instrum. **89**, 035114 (2018).
[29] A. Benítez, U. Santiago, J. E. Sanchez, and A. Ponce, Rev. Sci. Instrum. **89**, 013702 (2018).
[30] J. Zapata-Farfan, R. Contreras-Martínez, M. Rosete-Aguilar, J. Garduño-Mejía, P. Castro-Marín, O. G. Rodríguez-Herrera, N. C. Bruce, M. Ordóñez-Pérez, N. Qureshi, and G. Ascanio, Rev. Sci. Instrum. **90**, 045116 (2019).
[31] T. Cremers, N. Janssen, E. Sweers, and S. Y. T. van de Meerakker, Rev. Sci. Instrum. **90**, 013104 (2019).
[32] M. R. Hush, https://m-loop.readthedocs.io/en/latest/api/mloop.html, 2016.
[33] Throughout this article we have provided details of the commercial components that we have used, in order to assist the reader who wishes to duplicate our system. Components from other manufacturers may deliver similar, or better, performance.
[34] C. S. Adams and I. G. Hughes, *Optics f2f: From Fourier to Fresnel* (Oxford University Press, Oxford, UK, 2019).
[35] I. G. Hughes and T. P. A. Hase, *Measurements and Their Uncertainties* (Oxford University Press, Oxford, UK, 2010).
[36] D. R. Burt, C. E. Rasmussen, and M. van der Wilk, "Convergence of sparse variational inference in Gaussian processes Regression," J. Mach. Learn. Res. **21**, 131 (2020).
[37] The data presented in this paper are available from DRO http://doi.org/10.15128/r1qr46r0844.