Jalo Nousiainen

MODEL-BASED REINFORCEMENT LEARNING AND INVERSE PROBLEMS IN EXTREME ADAPTIVE OPTICS CONTROL

ACTA UNIVERSITATIS LAPPEENRANTAENSIS 1077

Jalo Nousiainen

MODEL-BASED REINFORCEMENT LEARNING AND INVERSE PROBLEMS IN EXTREME ADAPTIVE OPTICS CONTROL

Dissertation for the degree of Doctor of Philosophy to be presented with due permission for public examination and criticism in the Auditorium 1318 at the Lappeenranta-Lahti University of Technology LUT, Lappeenranta, Finland on the 26th of May 2023, at 10.00 a.m.

Acta Universitatis
Lappeenrantaensis 1077

Supervisors	Professor Tapio Helin LUT School of Engineering Science Lappeenranta-Lahti University of Technology LUT Finland
	Markus Kasper, PhD European Southern Observatory Germany
Reviewers	Associate Professor Andreas Hauptmann Department of Computational Mathematics The University of Oulu Finland
	Associate Professor Tim Morris Department of Physics Durham University United Kingdom
Opponent	Associate Professor Ville Kyrki Department of Electrical Engineering and Automation Aalto University Finland

ISBN 978-952-335-939-0
ISBN 978-952-335-940-6 (PDF)
ISSN 1456-4491 (Print)
ISSN 2814-5518 (Online)

Lappeenranta-Lahti University of Technology LUT
LUT University Press 2023

Abstract

Jalo Nousiainen
Model-based reinforcement learning and inverse problems in extreme adaptive optics control
Lappeenranta 2023
66 pages
Acta Universitatis Lappeenrantaensis 1077
Diss. Lappeenranta University of Technology
ISBN 978-952-335-939-0, ISBN 978-952-335-940-6 (PDF), ISSN 1456-4491 (Print), ISSN 2814-5518 (Online)

The field of exoplanet research is one of the most rapidly expanding research fields in modern astrophysics. In recent decades, astronomers have found most exoplanets via indirect techniques such as the transit and radial velocity method. The direct imaging technique called high contrast imaging (HCI) enables new ways to expand our knowledge of these exoplanets and exoplanetary systems. However, direct imaging of exoplanets is challenging due to the high contrast ratio and small angular separation from the host star. Thus, HCI detections, so far, are mostly limited to a few dozen young and luminous giant exoplanets.

The new generation of HCI instruments, under development, will push direct imaging into increasingly challenging areas, discovering and characterizing exoplanets dimmer and closer to their host start. The ultimate goal is direct imaging and characterization of potentially habitable exoplanets. On ground-based telescopes, HCI instruments are equipped with eXtreme Adaptive Optics (XAO) that correct the phase fluctuations caused by the atmosphere. With an optimized instrument design, the residuals left by XAO correction set the limitation of sensitivity; thus, minimizing the XAO residuals is a crucial objective for ground-based HCI. Further, most habitable exoplanets are located at small angular separations from their host stars, where current XAO control algorithms leave strong residuals of stellar light that could be suppressed with more advanced algorithms. This thesis explores novel data-driven control methods for XAO control that cope with crucial limitations of traditional control laws, such as temporal delay and calibration errors. Improvement in these potentially reduces the residual flux of stellar light in the coronagraphic point spread function and thus enables fainter observations closer to the host star.

We show that model-based RL is a promising XAO control approach that produces consistent results in numeric simulations and lab setups. The proposed methods suppress the temporal error, and photon noise compensates for misregistration and optical gain. It can also adapt to changing wind conditions in time scales of several seconds. Moreover, model-based RL manages the extreme time constraint of XAO control and, if well formulated, scales to ELT scale XAO.

Keywords: adaptive optics, high contrast imaging, reinforcement learning, inverse problems, robotics, and statistical machine learning

Acknowledgements

I want to express my sincere gratitude to all those who have supported me throughout my dissertation journey involving multiple scientific fields, universities and organizations, and cities.

Firstly, I would like to thank my supervisors, Tapio Helin and Markus Kasper, whose guidance, encouragement, and feedback have been instrumental in completing this dissertation. I would also like to thank Tapio for hiring me, an on-off math student with a variable studying record, for a research assistant and for sending me off on the first research visit to ESO headquarters in Garching after a few months of work; and Markus for offering partial funding for my dissertation process based on the modest preliminary results I presented on that specific visit. Those decisions set me off to a scientific career. I am also genuinely grateful for their patience and expertise, which have helped me refine my bold research visions into scientific publications and develop my research skills.

Secondly, I wish to thank my reviewers, Tim Morris and Andreas Hauptmann. Their insightful comments helped me to improve the thesis. I also want to thank Ville Kyrki, who kindly agreed to be my opponent.

Thirdly, I thank my colleague Chang Rajani for his collaboration with me, his support in machine learning and programming, and his direct and honest feedback. The main algorithms in the publications were developed and implemented jointly with him. The thesis would not be the same without his contribution. I would also like to thank my other colleagues. I thank Byron Engler and Sebastiaan Haffert for their support and work in implementing the algorithms for lab set-ups. I am looking forward to collaborating with them in the future. I also thank my collaborators, Tomi Krokberg, Jonatan Lehtonen, Christophe Vérinaud, and other co-authors.

I am grateful to all my dear colleagues and friends from the University of Helsinki; thanks for all the good memories at work, on conference trips, and outside work. I am also grateful to all my ESO colleagues for making me feel welcome and part of the AO group there. I also thank my colleagues from LUT University and Aalto University. I enjoyed working with you. I also thank Nuutti Hyvönen for offering me working space at Aalto University.

I thank my parents and family for their encouragement and support throughout my studies and my siblings for their friendship and co-support. I also thank my friends outside academia for providing some counterweight to academic work.

Finally, I thank Saara, my partner and best friend. I am forever grateful for her love, understanding, patience, and support throughout the thesis project: all the traveling, doubtful periods, and my absent-mindedness, as well as optimistic moments and long evening working sessions at home. With you, completing the thesis was much more manageable.

Jalo Nousiainen
March 2023
Lappeenranta, Finland

Contents

List of publications

Publication I
Nousiainen, J., Rajani, C., Kasper, M., & Helin, T. (2021). Adaptive optics control using model-based reinforcement learning. *Optics Express*, 29(10), 15327-15344.

Publication II
Nousiainen, J., Rajani, C., Kasper, M., Helin, T., Haffert, S. Y., Vérinaud, C., ... & Miller, K. (2022). Toward on-sky adaptive optics control using reinforcement learning-Model-based policy optimization for adaptive optics. *Astronomy & Astrophysics*, 664, A71.

Publication III
Nousiainen, J., Engler, B., Kasper, M., Helin, T., Heritier, C. T., & Rajani, C. (2022, August). Advances in model-based reinforcement learning for adaptive optics control. In *Adaptive Optics Systems VIII* (Vol. 12185, pp. 882-891). SPIE.

Publication IV
Krokberg, T., Nousiainen, J., Lehtonen, J., & Helin, T. (2022, August). FitAO: a Python-based platform for algorithmic development AO. In *Adaptive Optics Systems VIII* (Vol. 12185, pp. 1031-1037). SPIE.

As the first Author of papers I-II, Jalo Nousiainen was the corresponding author and had a leading role in writing the papers. Nousiainen conducted the numerical and lab experiments while the algorithms were developed and implemented jointly with Chang Rajani. In article III, Nousiainen wrote the paper and had a leading role in implementing the algorithm and conducting the experiments. In article IV, Nousiainen designed and supervised the work leading to the FitAO platform. He contributed substantially to the programming and had a major role in the writing process.

Nomenclature

Abbreviations

AO	Adaptive optics
CNN	Convolutional neural network
DM	Deformable mirror
DoF	Degrees of freedom
ELT	European Extremely Large Telescope
FoV	Field of view
FWHM	Full-width-at-half-maximum
HCI	High contrast imaging
KL	Karhunen-Loève
MBRL	Model-based reinforcement learning
MDP	Markov Decision Process
ML	Machine Learning
MLP	Multilayer perceptron
MPC	Model predictive control
MSE	Mean-square error
NCPA	Non-common path aberration
NN	Neural network
PCS	Planetary Camera and Spectrograph
POMDP	Partially Observed Markov Decision Process
PSF	Point spread function
PWFS	Pyramid wavefront sensor
RL	Reinforcement learning
RMS	Root-mean-squared
RTC	Real-time computer
S/N	Signal-to-noise ratio
SHS	Shack-Hartmann sensor
SL	Supervised Learning
SR	Strehl ratio
TSVD	Truncated singular value decomposition
WDH	Wind-drive halo
WFS	Wavefront sensor
XAO	Extreme adaptive optics

1 Introduction

Adaptive optics

Looking at the sky on a clear night, one may notice that stars seem to twinkle or change their brightness and position. In fact, almost every visible star shines steadily and still, and the turbulence in the atmosphere causes the twinkling effect. As the light from a distant astronomical object travels through the atmosphere, it gets distorted by an ever-changing mix of cold and warm air, in other words, by atmospheric turbulence. For astronomers, atmospheric turbulence means decreased image quality on ground-based telescopes, and for many decades it set the limits for the sharpness of images obtained.

Adaptive optics (AO) is a technique that aims to remove atmospheric distortions. The basic concept is to use a star (referred to as a guide star), for which we know the exact location, as a reference point to measure the distortion along the line of sight and an adaptive element (usually a deformable mirror) that can change its shape to compensate for the distortions caused by the atmosphere.

Proposed almost 70 years ago by Horace Babcock (1953), AO was successfully tested on-sky for the first time in 1974 (Hardy et al., 1974) and a bit more than a decade later for astronomical purposes in 1989 (Merkle et al., 1989). The results of Merkle et al. (1989) showed that AO could recover the diffraction limit of a 1.5-meter telescope. Since then, AO techniques and expectations have evolved in many directions, utilizing different astronomical observations. Some AO systems aim to produce a good correction on a wide field of view, some suitable corrections of very faint objects, and some good corrections in multiple directions simultaneously, while others aim for excellent corrections on a narrow field of view.

This thesis focuses on the development of so-called *eXtreme Adaptive Optics* (XAO) systems. ExAO systems are adaptive optics (AO) systems specifically designed to provide excellent wavefront correction on relatively bright natural guide stars on small angular separations (close to the guide star). They typically operate at a higher speed and have actuators than general-purpose AO systems. XAO relies on a single on-axis star for wavefront sensing and does not address the anisoplanatism effects that restrict the AO-corrected field of view. Currently, these systems are found in instruments dedicated to direct exoplanet imaging, such as the Gemini Planet Imager (Macintosh et al., 2014) on the Gemini South telescope and the SPHERE (Spectro-Polarimetric High-contrast Exoplanet REsearch, Fusco et al., 2006; Beuzit et al., 2010) instrument on the European Southern Observatory's Very Large Telescope and newer experimental instruments MagAO-X (Magellan Adaptive Optics eXtreme system, Males et al., 2018) and SCExAO (Subaru Coronagraphic Extreme Adaptive Optics, Jovanovic et al., 2015).

What are exoplanets and exo-earths?

Most of us are familiar with the planets that orbit our Sun. However, our galaxy, the Milky Way, contains about 400 billion stars, our Sun among them. Moreover, like our Sun, many have not just one but a whole system of planets orbiting them. The planets orbiting these other stars are called exoplanets, They come in a wide variety of sizes and with a wide

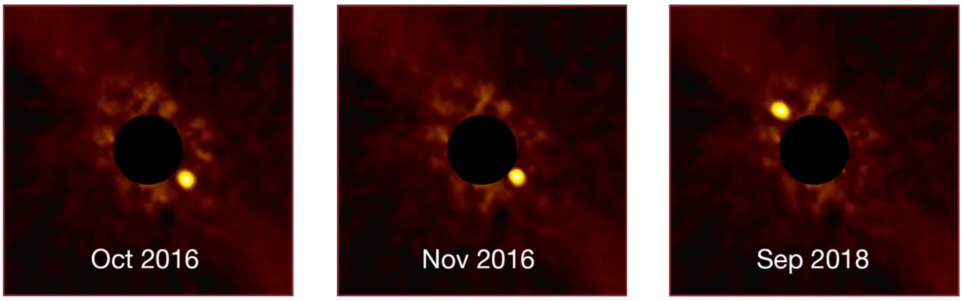


Figure 1.1: A time-lapse picture series of exoplanet Beta Pictoris b. These stunning pictures, showing Beta Pictoris b's orbit around its host star, were caught with SPHERE, an instrument dedicated to direct exoplanet imaging. Sixty-three light-years away, planet Beta Pictoris b orbits its host star at a distance similar to that between the Sun and Saturn. The host star is blocked with a black digital mask. So far, it is the most closely orbiting exoplanet ever captured by the direct imaging technique. The development of the new generation of exoplanet imaging instruments will push direct imaging into increasingly challenging areas – discovering and characterizing exoplanets dimmer and closer to their host start.

variety of other properties, from gas giants like Jupiter to smaller rocky planets like Earth, and from melting hot to freezing cold.

The planets of size and mass almost equal to the Earth's and located in the habitable zone around a star are often called exo-earths. These planets are potential hosts for life (as we know it) since they are located in an area where the temperature is just right for liquid water to exist on their surface. Studying these planets is especially interesting as it may result in finding unmistakable signs of current life on a planet beyond Earth.

From indirect observations to direct exoplanet imaging

During the last decade, NASA's Kepler mission[1] has identified over 3000 confirmed exoplanets through an indirect technique called the transit method. Moreover, most of the more than 5,000 exoplanets confirmed have been found by indirect methods, such as the transit or the radial velocity method. The transit method measures the dimming of a star that has a planet pass in front of it, and the radial velocity method monitors the spectrum of a star for the telltale signs of a planet's gravitational pull on its star causing the light to subtly Doppler shift.

The indirect methods do not produce a direct image of the planet but give an indirect indication of the planet's existence. In contrast, a direct imaging technique called *High contrast imaging* (HCI) aims to separate the exoplanet light from the stellar light optically, producing a direct image of the planet. Figure 1.1 shows an HCI observation, that

[1] Exoplanet Orbit Database: http://exoplanets.org/

is, a direct image, of an exoplanet Beta Pictoris b. As the glaring stellar light is usually the dominating source of measurement noise, it dramatically increases the signal-to-noise ratio (S/N) over that provided by indirect methods. So far, HCI detections are mostly limited to a few tens of very young and luminous giant exoplanets (e.g., Marois et al., 2010; Lagrange et al., 2009; Macintosh et al., 2015). The main reason behind the small number of HCI detections is that they are exceptionally challenging: the planets produce very little light of their own and they are at an enormous distance from us yet reasonably close to their host star. Consequently, the planets are lost in the blinding glare of their parent stars, and the direct imaging of exo-earths remains unfeasible for even the most advanced existing HCI instruments.

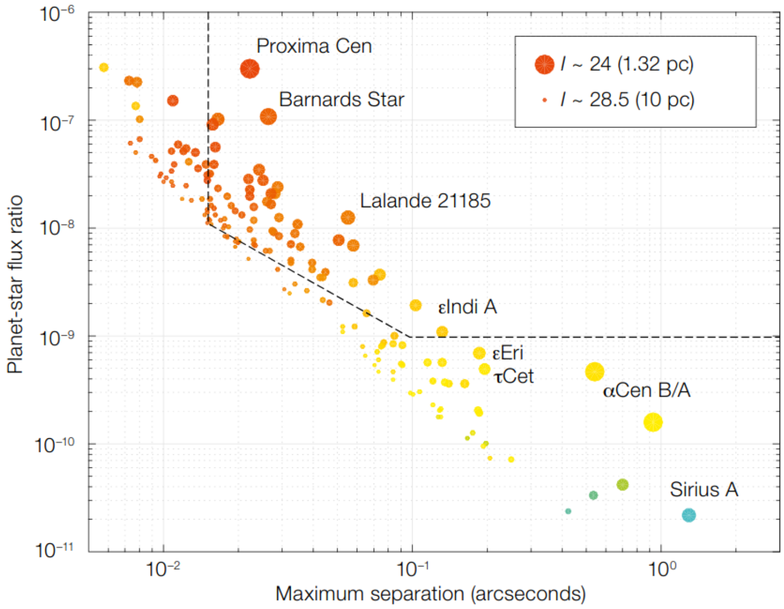


Figure 1.2: I-band flux ratio between hypothetical exo-earths and parent stars within ten (10) parsec (observable from the ELT construction site) as a function of angular separation. The symbol size reveals the planet's apparent brightness, and the colors indicate the stellar spectral type (red: M-stars, yellow: solar-type stars). The approximate contrast boundaries for PCS, i.e., the ability to distinguish the reflected light from the planet, are shown as a dotted line (Kasper et al., 2020).

The Planetary Camera and Spectrograph for the Extremely Large Telescope

The European Extremely Large Telescope (ELT) is an observatory currently under construction. When completed, it will be the world's largest optical/near-infrared telescope, and it is planned to enable fundamental contributions to astronomy and cosmology. It will facilitate multiple large science instruments for different kinds of observations, from studying the very first galaxies in the so-called ”Dark Ages” to tracking down exo-earths around nearby stars.

The Planetary Camera and Spectrograph (PCS, Kasper et al., 2020) for the ELT will be dedicated to detecting and characterizing nearby exoplanets with sizes from sub-Neptune to Earth-size around the nearby planetary systems. Figure 1.2 illustrates the approximate contrast boundaries of PCS along with some potential targets as a function of angular separations. To achieve this ambitious goal, PCS combines XAO, coronagraphy, and spectroscopy. If the goals are met, PCS will allow us to take direct images and look for biosignatures such as molecular oxygen in the exoplanets' atmospheres.

For PCS, the performance of the XAO system plays a crucial role and is a technology still requiring significant research and development (R&D). It is critically important to minimize the photon noise introduced by stellar light scattered to the position of the nearby planet, which is the primary noise source for ground-based exoplanet detection. This thesis contributes to the PCS's R&D activities at the European Southern Observatory (ESO).

Development of XAO control methods

The most interesting objects for direct imaging, such as exo-earths, are often located at very small angular separations from their host stars. At small angular separations, one of the limiting factors is the servo-lag of the system, which could be compensated for by advanced control algorithms. Therefore, for future (and current) HCI instruments, new control approaches and techniques have the potential to offer significant performance gains (Guyon, 2005). Unsurprisingly, advanced XAO control methods have gained significant attention in the research field of HCI instrumentation in recent years. These methods include the Kalman filter-based linear controllers (Kulcsár et al., 2006; Paschall and Anderson, 1993; Gray and Le Roux, 2012; Conan et al., 2011; Correia et al., 2010b,a, 2017), sometimes combined with machine learning for system identification (Sinquin et al., 2020). Other methods vary from linear filters to filters operating on single modes, such as Fourier or Zernike modes (Guyon and Males, 2017; Poyneer et al., 2007; Dessenne et al., 1998; van Kooten et al., 2017, 2019), to neural network approaches (Swanson et al., 2018; Sun et al., 2017; Liu et al., 2019; Wong et al., 2021). Predictive control methods have also been studied in a closed-loop configuration. Males and Guyon (2018) address a closed-loop predictive control's impact on the post-coronagraphic contrast with a semi-analytic framework. Swanson et al. (2021) studied closed-loop predictive control with NNs via supervised learning, where a NN is learned to compensate for the temporal error. Some methods have also been tested on-sky (e.g., van Kooten et al., 2022). More recently, remarkable progress has been achieved with fully data-driven control methods that, in addition to temporal prediction, add the control signals to the learned model to account for closed loop dynamics (Pou et al., 2022; Landman et al., 2020, 2021; Haffert et al., 2021a,b).

This thesis aims to explore novel data-driven control methods for XAO control for predictive self-calibrating control. In particular, it focuses on *Reinforcement Learning*

(RL) methods for AO from the perspective of studying the potential benefits of these approaches and their practical implementations. XAO is a closed-loop control system where the system's state is observed via indirect, ill-posed measurement. More precisely, the atmospheric turbulence, its evolution model, and the DM surface are not observed directly but through a sensor measurement that gives indirect data. Moreover, the system dynamics are not known a priori, but they must be approximated from the data, either from the observation of some external sensor. RL methods learn control solely from the interaction with the system and are, therefore, insensitive to the common pitfalls of pseudo-open-loop predictive controllers, such as the optical gain effect, misregistration, and temporal jitter/-timing errors. The framework of RL methods for XAO control positions the thesis at the interface of three research fields: *astronomical instrumentation, inverse problems*, and *reinforcement learning*.

2 Fundamentals

This section explains the fundamentals of physics behind astronomical telescopes and adaptive optics. Along with mathematical notation, these are explained through numerically simulated examples. All XAO simulations in the introduction are performed with COMPASS simulator (Ferreira et al., 2018), while the science camera and coronagraph are simulated with HciPy toolbox (Por et al., 2018).

As this introduction aims only to give a short description of AO from the perspective of the scientific results of the thesis, we encourage the interested reader to peruse textbooks by, for example, Roddier (1999), Hardy (1998), Hickson (2008) and Roggemann et al. (1996) for more detail. We also encourage the reader to look through Ellerbroek and Vogel (2009) for interesting inverse problems in AO that are not discussed in this thesis.

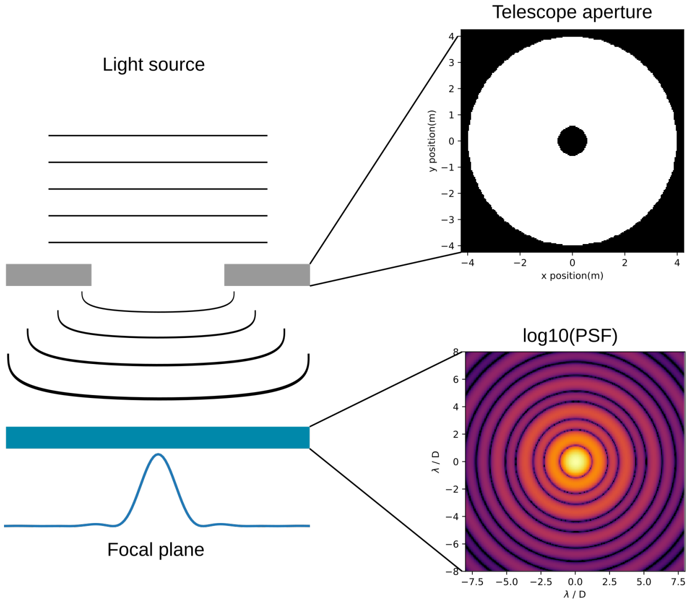


Figure 2.1: Illustration of diffraction effect and the Fraunhofer approximation. The light from a faraway star (a point source at infinity) produces a planar wavefront. The telescope aperture causes the light to diffract and form a diffraction pattern (the Airy disk) on the focal plane

2.1 Diffraction limit

The purpose of a telescope is to collect light from a distant source and focus it as an image on the science camera, that is, the image plane. Generally, the shape of the primary mirror defines the entrance aperture that restricts the rays that reach the image. By the principles of physics, the aperture causes light waves to spread out and interfere with one another and, consequently, small details on the image plane to blur. This effect, called diffraction, sets a theoretical limit for the telescope's image quality.

The most natural way to explain diffraction for astronomical imaging is by Fourier optics (Goodman, 2005). All astronomical objects are very distant compared to the telescope aperture; hence, it is common to use the so-called Fraunhofer far-field approximations to model the diffraction pattern. The point spread function (PSF) $s_0(u, v)$ of an ideal telescope (i.e., an image of a faraway monochromatic point source) is then described by a Fourier transform of the aperture shape:

$$s_0(u,v) = |\mathcal{F}\{\chi_\Omega\}(u,v)|^2 = \left| \iint \chi_\Omega(x,y) e^{-2i\pi(xu+yv)} \, \mathrm{d}x \, \mathrm{d}y \right|^2 , \tag{2.1}$$

where the $\chi_\Omega(x, y)$ is a characteristic function of the telescope aperture and (u, v) spatial frequencies scaled by λ/D. The Fourier spatial frequencies (u, v) represent the spatial location at the image plane, that is, the science camera, and the value $s_0(u, v)$ represents the relative intensity of the light on it.

Due to the Fourier optics (Goodman, 2005), the image I of an arbitrary astronomical object f (e.g., an exoplanet and host star or a galaxy) can now be described as the convolution of the observed light intensity with the *diffraction limited* PSF s_0, that is,

$$I(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s_0(u-u', v-v') f(u,v) \, \mathrm{d}u' \, \mathrm{d}v' \tag{2.2}$$

The convolution operator blurs the image; hence, the images' high spatial frequencies are dampened. A telescope's ability to differentiate details is often defined by the so-called *Rayleigh criterion*. It states that two images are just resolvable when the center of the diffraction pattern of one is directly over the first minimum of the diffraction pattern of the other; see Figure 2.2. The diffraction pattern's first minimum is approximately at point $1.22\lambda/D$. Two points are just resolvable if this angle separates them.

The diffraction-limited angular resolution (1.22 λ/D) of the telescope depends on the imaging wavelength λ and the telescope diameter D. Thus, at a given wavelength, resolving smaller angular separation needs bigger telescopes. For an 8-meter telescope, imaging at wavelength 850 nm (I-band), 1.22 λ/D is approximately 25 milliarcseconds, which is enough spatial resolution to resolve many potentially habitable planets around nearby stars. However, on ground-based telescopes, the difficulty of exoplanet imaging lies in the enormous flux ratio (contrast) between the planet and the host star (see Figure 1.2), the atmospheric turbulence, and in the telescope's ability to collect enough photons in reasonable observing time.

2.2 The effect of phase aberrations

The diffraction pattern gives a theoretical limit to the telescope's image quality, but in the real world, any optical system will introduce aberrations to the incoming wavefront. We denote the aberrated incoming electromagnetic field (the wavefront) by $\psi : \mathbb{R}^2 \to \mathbb{C}$, defined by:

$$\psi(x,y) = A_\Omega(x,y) e^{-i\phi(x,y)}, \tag{2.3}$$

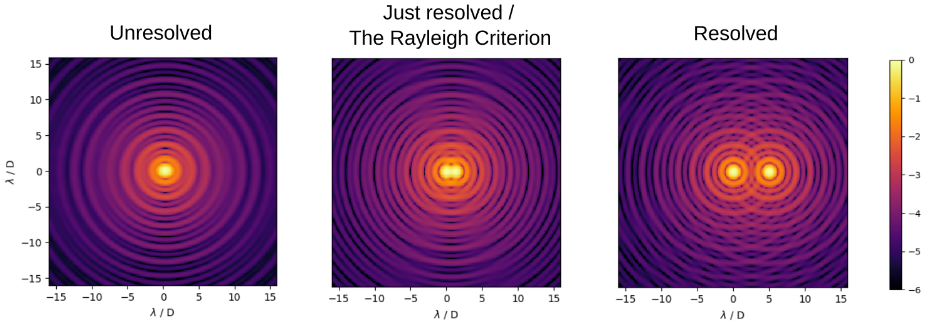


Figure 2.2: The Rayleigh criterion. The focal plane image of two point sources at three different separations: 0.5 λ/D (unresolved), 1.22 λ/D (just resolved) and 5 λ/D (resolved). The color indicates the PSF relative brightness compared to the peak intensity in the middle, i.e., the contrast in a logarithmic scale.

where $A_\Omega(x, y)$ is the amplitude over the pupil aperture and $\phi(x, y)$ the phase aberrations. Neglecting the amplitude variation and normalizing the amplitude, the function A is simply the characteristic function of the pupil χ_Ω, and the Fraunhofer approximation of a monochromatic point source (PSF) is now given by,

$$s_\phi(u, v) = |\mathcal{F}\{\chi_\Omega e^{i\phi}\}(u, v)|^2. \tag{2.4}$$

The image of the object f on the focal plane would be again convolved, but this time with the aberrated PSF s_ϕ. For example, let us look at the simple static planar tip and tilt aberration in the phase, that is, $\phi(x, y) = 2\pi(\theta_1 x + \theta_2 y)$. Now solving (2.4), with tip-tilted ϕ, yields

$$\begin{aligned} s_\phi(u, v) &= \left| \iint \chi_\Omega(x, y) e^{-2\pi(x(u-\theta_1)+y(v-\theta_2))} \,\mathrm{d}x\,\mathrm{d}y \right|^2 \\ &= s_0(u - \theta_1, v - \theta_2). \end{aligned} \tag{2.5}$$

We note that a tip-tilt aberration causes a linear shift in the image of an object. If this aberration evolved during the exposure, the image would eventually be blurred beyond the diffraction pattern.

2.3 Strehl ratio

The *Strehl ratio* (SR) is a common way to measure the quality of the PSF. It is defined as the ratio between the peak intensity of observed PSF at the image plane and the theoretical diffraction-limited PSF, that is,

$$\mathrm{SR} = \frac{s_\phi(0, 0)}{s_0(0, 0)}. \tag{2.6}$$

The SR is connected to the variance of the phase aberration ϕ via two Marechal's approximations (Tyson and Frazier, 2022):

$$\frac{s_\phi(0,0)}{s_0(0,0)} \approx e^{\hat{\sigma}^2_\phi}, \tag{2.7}$$

where $\hat{\sigma}^2_\phi$ is the ensemble variance (over time) of the phase errors, and

$$\frac{s_\phi(0,0)}{s_0(0,0)} \approx 1 - \sigma^2_\phi, \tag{2.8}$$

where σ^2_ϕ is the phase variance over the pupil on a given time instance. The approximation in Equation (2.8) follows from the Taylor series expansion (about $\phi = 0$) of the exponential in Equation (2.5) and is, hence, only valid for small phase aberration, i.e., high SR ($> 80\%$). The approximation (2.7) is valid for larger phase errors, but it is used exclusively for long-exposure SR. Consequently, for moderate and high SRs, minimizing the wavefront error variance, i.e., the mean-square error (MSE), maximizes the SR.

2.4 Imaging through turbulence

On ground-based telescopes, the atmosphere is the primary source of phase aberrations in the incoming light. Atmospheric turbulence is the random mix of air of different temperatures in a constant motion driven by the wind. It introduces spatial and temporal variations in the refractive index of air and, thus, in the optical path length of the stellar light. These variations are governed by the Kolmogorov-Obukhov turbulence law.

In the early 1940s, Kolmogorov and Obukhov developed a statistical model to describe turbulent air motion (Kolmogorov, 1991). The model is limited to a spatial frequency band, called the inertial range, bounded by the outer and inner scales. Kolmogorov and Obukhov supposed that the turbulence is initialized in the outer scale L_0 and progresses systematically to smaller scales as large whirls transfer energy to small whirls. When this turbulence energy reaches the inner scale l_0, it dissipates to heat energy due to viscous friction. Kolmogorov and Obukhov suggested that inside the inertial range, the turbulence field (fluctuations in the refractive index) can be modeled as a stationary and isotropic Gaussian random field at any fixed time instance. Interested readers may refer to Lukin (1995) for a comprehensive theoretical examination of adaptive optics-related turbulence phenomena.

The vertical profile of the turbulence is often modeled as a finite set of thin, statistically independent turbulent layers at a given height and with a given turbulence strength. Moreover, compared to changes depending on wind speed, the turbulence pattern changes are relatively slow. Hence a good approximation of the time evolution is reached by Taylor's frozen flow hypotheses. Each turbulent layer is modeled as a thin static 'frozen' layer sliding over the telescope with an individual wind speed and direction.

A good approximation for most astronomical AO applications is the neglection of amplitude variations (or scintillation), and the turbulence is assumed only to induce phase variations to the wavefront ψ (Davies and Kasper, 2012). For HCI systems (introduced in

Section 3) the amplitude variations introduce an error term for the contrast performance (Guyon, 2005). However, correcting for amplitude variations is beyond the scope of this thesis and is not modeled by the numerical simulation. The phase errors ϕ are obtained by integrating over the layers along the optical path to the telescope. For a more detailed description of wave propagation through the atmosphere, the reader can refer to, e.g., Tatarski (2016); Roddier (1999); Tyson and Frazier (2022). By using a subtle variation on the original Kolmogorov model, phase variations after each thin layer ϕ, modeled as a two-dimensional Gaussian process ($\mathcal{GP}$) follow so-called von Kàrmàn statistics, that is,

$$\phi(x, y) \sim \mathcal{GP}(0, C_\phi((x, y), (x', y'))), \tag{2.9}$$

where C_ϕ only depends on the Euclidean distance between locations (x, y) and (x', y') and the *power spectral density* (PSD), that is, the Fourier transform of the covariance function is given by

$$\hat{C}_\phi(\boldsymbol{\kappa}) = a(|\boldsymbol{\kappa}| + 1/L_0^2)^{-11/6}. \tag{2.10}$$

Here $\boldsymbol{\kappa}$ is the spatial frequency, L_0 is the outer scale of the layer, and a is a constant defining the turbulence strength. The PSD expresses the amount of turbulent energy at a given spatial frequency. In general, one could argue that the Kolmogorov spectrum (infinite outer scale) is not physical and that a suitable typical largest spatial scale L_0 has to be introduced as done by the von Kàrmàn spectrum. Given (2.10), the corresponding covariance function belongs to the family of Whittle-Matèrn random fields, with smoothness parameter 5/6 (Doelman, 2020). We note here that the smoothness parameter $5/6$ refers to the smoothest of the covariance function, not the exponent $(-11/6)$ in PSD.

In the near-field approximation, the final cumulative optical path aberrations ϕ_{total} (henceforth referred to simply as ϕ) is a sum of aberrations along the line of sight. That is, simply the sum of phase variations after each thin layer:

$$\phi(x, y) = \sum_{l=1}^{L} c^l \phi_l(x, y), \tag{2.11}$$

where L is the total number of layers and ϕ_l phase aberrations after the corresponding layer, the collection of relative strengths $[c^1, c^2, \cdots, c^L]$ at layers is called the discrete C_N^2 profile (a typical notation in AO literature).

2.5 Atmospheric turbulence parameters

Regarding HCI, the two most essential atmospheric turbulence quantities are the *Fried parameter* and the *coherence time*. The Fried parameter (Fried, 1966) was introduced to describe the magnitude of the atmospheric turbulence effect on the telescope's image quality. It is the diameter of the circular aperture over which the root-mean-squared

(RMS) wavefront phase aberration equals one radian. That is given by

$$r_0 = \left(0.423 \cdot \left(\frac{2\pi}{\lambda} \right)^2 (\sec \zeta) \int_0^h C_n^2(z) dz \right)^{-3/5}, \tag{2.12}$$

where λ is the wavelength and ζ is the zenith angle, that is, the view direction. The refractive index structure constant C_n^2 describes the strength of the turbulence as a function of the altitude $0 \leq z \leq h$ (a discrete vector in the layered model).

The Fried parameter can also be interpreted as the aperture size for which the diffraction effect and the effect of turbulence cause an equivalent decrease in image resolution. That is, phase aberrations smaller than one radian RMS mean diffraction-limited resolution. Moreover, the r_0 is closely related to so-called astronomical seeing, the full-width-at-half-maximum (FWHM) of the turbulence-blurred PSF, usually measured in arcseconds ($''$). For example, in $1''$ seeing the Fried parameter is in order 10-cm at visible wavelengths.

Similar to the spatial variation of the turbulence, one can derive a corresponding parameter for the temporal evolution of the turbulence. The coherence time, also called the Greenwood time delay, is given by

$$\tau_0 = 6.88^{-3/5} \frac{r_0}{\boldsymbol{v}}, \tag{2.13}$$

where r_0 is the Fried parameter and $\boldsymbol{v}$ the average wind speed. The coherence time τ_0 gives the time interval over which the phase aberration changes approximately one radian of RMS.

The Fried parameter and coherence time set complexity requirements for the AO system. The Fried parameter determines the spatial resolution requirement of phase correction, that is, the number of spatial frequencies the AO system has to be able to correct to reach an adequate Strehl ratio. The coherence length determines the system's temporal correction bandwidth, ultimately defining the AO system's loop speed. Typically the astronomical seeing is reported at $\lambda = 500$nm (visual light), and the common median seeing for a good astronomical site is from $0.6''$ to $0.7''$, corresponding to a Fried parameter of 14.4cm to 16.8cm. The corresponding coherence time for a typical average wind speed of 10m/s is approximately 5ms. Both, r_0 and τ_0 are proportional to $\lambda^{6/5}$, i.e., r_0 is typically around 40cm at $\lambda = 1.6$µ. Figure 2.3 shows the effect of atmospheric turbulence on the PSF under typical seeing conditions.

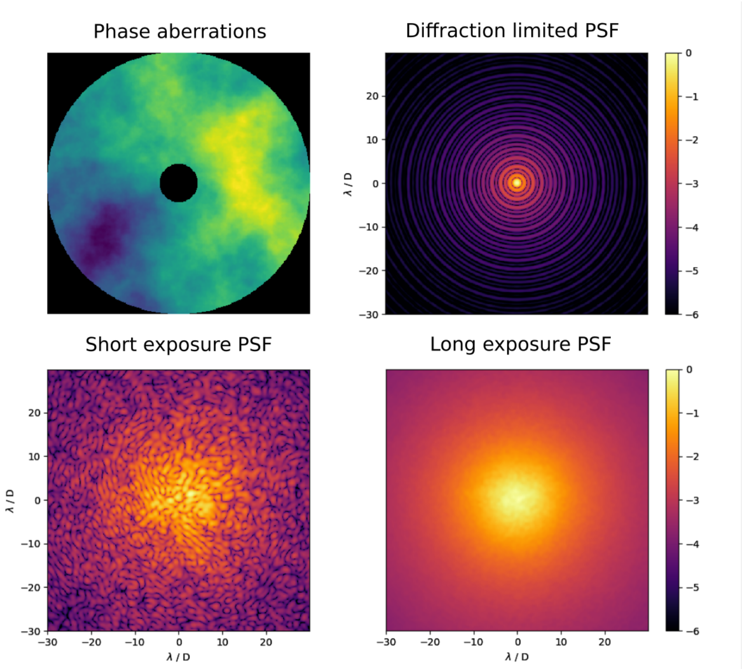


Figure 2.3: The effect of atmospheric turbulence on the PSF at 1.6 μm with $r_0 = 16$cm at 500nm. *Upper left*: Phase aberrations at a single time instance (frame). *Upper right*: the diffraction-limited PSF for a circular telescope aperture with a small central obstruction (e.g., the VLT). *Lower row images*: the corresponding PSF of the frame and the log exposure PSF over 6000 frames. In a single time instance, the PSF is broken up into many individual speckles. As phase aberrations change, the speckle patterns blur the PSF, thus reducing the resolving power.

3 High contrast imaging

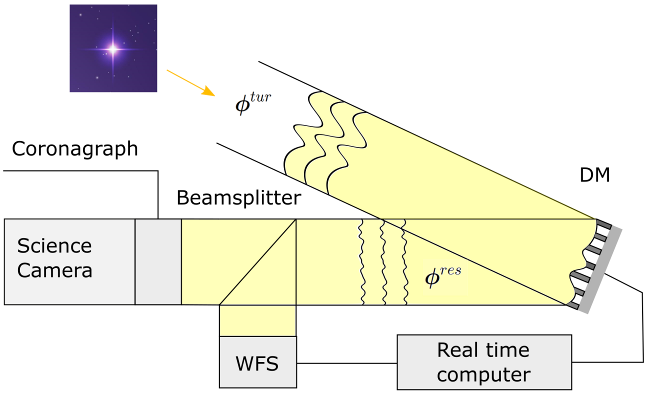


Figure 3.1: Illustration of simplified HCI system.

This section gives an overview of HCI systems. It introduces the most relevant components with emphasis on the XAO control. First, we introduce the XAO components, namely the *wavefront sensor* (WFS) , the *deformable mirror* (DM), and the control problem, where we outline a standard control law called integrator and the related calibration process. Then, we provide a brief introduction to coronagraphy and introduce the essential noise terms and the evaluation metrics used in the thesis.

An overview of a simplified HCI instrument is given in Figure 3.1. It comprises a single WFS, a single DM, and a *real-time computer* (RTC), which together compensate for the wavefront aberration. A coronagraph then suppresses the host star's glare to make the much fainter companion visible. However, modern XAO systems are usually much more complex than this simplified design and may involve multiple DMs, WFSs, and sensing techniques (Guyon, 2005, 2018; Males et al., 2018).

All HCI systems use a single guide star (the host star for the potential planet) as the reference source because the targets are nearby and usually bright enough for a WFS working at optical or near-infrared wavelengths. The WFS, which measures deviations from a flat wavefront, is set downstream from the DM. This configuration allows a closed-loop architecture – the WFS observes changes in the DM shape. More precisely, the DM corrects the incoming light ϕ_t^{tur} at the timestep t. After this correction, the WFS measures the residual wavefront ϕ_t^{res}. After receiving the wavefront sensor measurement, the RTC calculates a set of control voltages and sends the commands to DM; see Figure 3.1. Further, the AO control loop inherits a temporal delay. The delay consists of measurement delay introduced by the WFS detector integration and control delay consisting of detector readout, computations of the correction signal, and its application to the DM. These add up to a typical total delay of two update steps of the AO system running at the maximum speed of the WFS camera when the camera readout already takes about one update step (or frame).

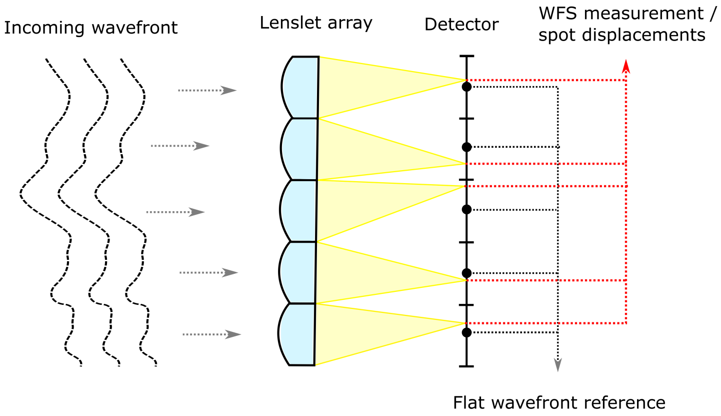


Figure 3.2: Illustration SHS. The SHS consists of a lenslet array and a detector. Flat wavefront reference points are the measurement of a non-aberrated wavefront. The SHS measures the focus points' displacement from the flat reference points.

3.1 Wavefront sensing

The WFS is one of the essential elements of an AO system. The function of the WFS is to measure the spatial shape of the phase including the DM corrections, that is, residual phase screen ϕ_t^{res}. There are different types of WFSs, and in this work, we introduce the two most common ones – the pyramid wavefront sensor (PWFS) and the Shack-Hartmann sensor (SHS).

Shack-Hartmann wavefront sensor

The SHS directs the incoming wavefront from the guide star to a lenslet array of small identical lenslets. Each of these lenslets then forms an image of the star onto the image plane. If the incoming wavefront is a plane wave, the images form a perfect grid on the image plane; as soon as the wavefront is perturbed, the images get displaced as in Figure 3.2. The displacement of an image is proportional to the average wavefront slope over the area of the corresponding lens (sometimes referred to as sub-aperture). These displacements from the flat wavefront focus points are the SHS measurements.

A simple model for SHS measurements is given by the average phase gradients over each lenslet (or sub-aperture):

$$w_x(i,j) = \frac{1}{|A_{(i,j)}|} \int\limits_{A_{(i,j)}} \partial_x \phi(\mathbf{r}) d\mathbf{r} \tag{3.1}$$

and

$$w_y(i,j) = \frac{1}{|A_{(i,j)}|} \int\limits_{A_{(i,j)}} \partial_y \phi(\mathbf{r}) d\mathbf{r}, \tag{3.2}$$

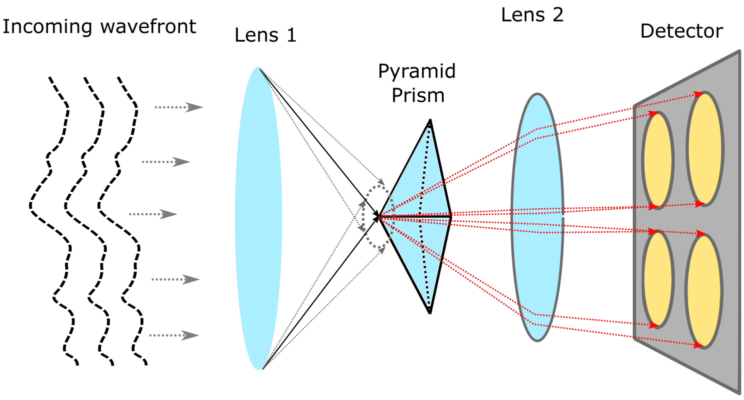


Figure 3.3: The working principle of PWFS. The focal plane PSF is directed to a four-sided pyramid, with or without spatial modulation (circular motion around the tip). Then, the light propagates to four different intensity images. The spatial shape of the phase can be recovered from the intensity images.

where $A_{(i,j)}$ is the sub-aperture surface at the position (i, j) and $|I|$ the total number of sub-apertures. The collection of w_x, w_y at all possible locations is denoted by a measurement vector $\boldsymbol{w}$.

This simple mathematical model is entirely linear. However, in reality, each lenslet has a corresponding detector area (or field of view, FoV) with a certain number of pixels and, due to the residual phase errors, the image on the detector is not ideal. Therefore, the SHS sensor's linearity depends on the detector's available FoV, the pixel size, and the algorithm used to retrieve the spot displacement (e.g., Basden et al., 2011). However, in the closed-loop regime of XAO operation, the SHS behaves very similarly to the average gradient model.

Pyramid wavefront sensor

The PWFS (Ragazzoni, 1996; Ragazzoni and Farinato, 1999) is a Fourier Filtering type of WFS, that operates in the focal plane (Fauvarque et al., 2016). In PWFS sensing, the electric field of the incoming wavefront is directed to a transparent four-sided pyramid prism. The prism is located in the focal plane of an optical system and the incoming light is usually modulated around the tip of the pyramid to various degrees; see Figure 3.3. This four-sided pyramid divides the incoming light in four different directions, and most of the light is propagated to four intensity images on the PWFS detector. Due to the slightly different optical paths of the light, the intensity fields differ from each other. These differences are then used as the data for recovering the disturbances in the incoming phase screen.

The amount of modulation on the pyramid's tip alters the properties of the PWFS. When strong modulation is applied, the PWFS closely represents the SHS sensing. On the other hand, small modulation, or the extreme case of zero modulation, delivers better sensitivity but compromises the dynamic range and linearity. However, the modulation amplitude can be easily adjusted to given imaging conditions (Guyon, 2005). This flexibility and high sensitivity make the PWFS a better choice for XAO. Newer XAO systems, such as SCExAO and MagAO-X, have adopted this relatively new wavefront sensing concept.

Let us consider the mathematical model of a non-modulated standard four-sided PWFS. The modulated PWFS would simply employ the following model along the circular modulation path. The PWFS can be viewed as a generalization of the Foucault knife-edge test (Ragazzoni, 1996), and it can be modeled as a spatial Fourier filter that introduces specific phase changes according to the shape of the prism (Fauvarque et al., 2017).

We denote the incoming phase screen by $\phi : \mathbb{R}^2 \to \mathbb{R}$ and the corresponding incoming electromagnetic field $\psi : \mathbb{R}^2 \to \mathbb{C}$ by

$$\psi(x,y) = \sqrt{n}\chi_\Omega(x,y)\exp[-i\phi(x,y)], \tag{3.3}$$

where n is the spatially averaged flux, and χ is the characteristic function of pupil. The electric field is then directed to a transparent four-sided pyramid prism. We can then estimate the intensity on the detector by using diffraction theory and the Fraunhofer approximations, that is,

$$I(x,y) = \left|\psi(x,y) * (\mathcal{F}^{-1}\{\text{OTF}_{pyr}\})(x,y)\right|^2, \tag{3.4}$$

where $\mathcal{F}^{-1}\{\text{OTF}_{pyr}\}$ is the pupil plane point spread functions of the glass pyramid (PSF$_{pyr}$). The PSF$_{pyr}$ is defined as the inverse Fourier transform of its optical transfer function OTF$_{pyr}$ that is characterized by the shape of the pyramid prism (Heritier, 2019; Shatokhina et al., 2020).

A four-sided pyramid divides the incoming light in four different directions, and most of the light is propagated to four intensity images on the PWFS detector. We denote these pupil images, i.e., intensity fields, by I_1, I_2, I_3, and I_4. Due to the slightly different optical paths of the light, the intensity fields differ from each other. As discussed before, we use these differences as the data for recovering the disturbances in the incoming phase screen.

Commonly, PWFS data, the intensity fields, are processed to so-called slopes w_x, w_y that correlate positively to actual gradients fields of the phase screen. In the articles of this thesis, we follow the approach of Vérinaud (2004), where the slopes are normalized with the global intensity,

$$w_x(x,y) = \frac{I_1(x,y) - I_2(x,y) + I_4(x,y) - I_3(x,y)}{I_{glob}} \tag{3.5}$$

$$w_y(x,y) = \frac{I_1(x,y) - I_4(x,y) + I_2(x,y) - I_3(x,y)}{I_{glob}} \tag{3.6}$$

In practice, we receive a vector $\boldsymbol{w}$ that is a collection of the measurements w_x, w_y at all possible locations x, y.

Currently, most wavefront reconstruction algorithms, discussed in Sections 3.3, utilize a linearization of the PWFS model, inducing a trade-off between sensitivity and robustness (modulated PWFS vs. non-modulated PWFS). Further, the PWFS is a *diffraction-limited wavefront sensor*, meaning that its sensitivity varies depending on both the seeing conditions and the level of AO correction itself (Korkiakoski et al., 2008). How to deal with these properties is still an active field of research, where different (outside this thesis) machine learning methods are also studied (e.g., Landman and Haffert, 2020).

3.2 Deformable mirror

Another critical component of AO systems is the DM which executes the required wavefront correction, taking the shape of the phase aberrations but with half the amplitude. A DM consists of a highly reflective cover that can be controlled from below with actuators (typically in a Cartesian grid). A DM has four key parameters: stroke, response time, actuator spacing, and the number of actuators. The requirements for actuator spacing and the response time are governed by the atmospheric parameters r_0, τ_0, and the science objective of the instrument, while the stroke (the difference between the highest and lowest actuator position) and the number of actuators, also referred to as DMs Degrees of Freedom (DoF) are scaled according to the aperture size.

Different DM designs are used to reach the astronomical AO requirements for these parameters, combining different low-order and high-order DM as well as developing new DM technologies. To this end, three leading DM technologies are the piezo-stack DMs that use stacks of piezo-ceramic disks for actuation, the adaptive secondary mirrors, and the micro-electromechanical system (MEOMS) devices. The reader can refer to, for example, Madec (2012); Roddier (1999) for a more detailed description of the different DM technologies.

We use here a relatively simple model for the piezo-stack type of DM. The DM actuators are positioned into a cartesian grid. Each actuator is controlled by control voltage defining the height of the actuator, and the deformations corresponding to a single actuator's actuation define the mirror's influence functions. The influence functions of the mirror are simulated with Gaussian bell curves, and the mechanical coupling gives the curve's height on the neighboring actuator; see Figure 3.4. In numerical simulations, we assume that the DM has enough stroke for the given imaging conditions, and DM saturation does not happen.

Modal basis

Instead of considering DM commands as individual zonal commands on the actuators, it is often more convenient to consider the commands on an orthogonal modal basis. Each command in the modal basis is a set of control voltages for all actuators in such a way that they together form a specific shape on the DM surface that is orthogonal to all the other modes; see Figure 3.4. The modal approach is advantageous for various reasons depending on the basis used. For example, the Zernike polynomials (Noll, 1976) are well suited to describe the optical aberrations commonly seen in optics; on the other hand, the Fourier basis gives a direct tool to analyze the spatial frequency content of the DM shape.

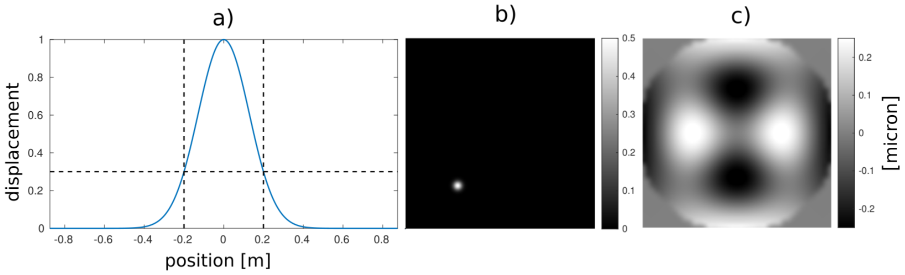


Figure 3.4: Gaussian DM influence function with 0.3 mechanical coupling and modal command. The simulated DM has 41 x 41 actuators across an 8-meter aperture (20cm spacing). *Panel a*: cross-section of the DM influence function. The dashed black lines mark the position of the neighboring actuators. *Panel b*: A single actuator pushed on the DM. *Panel c*: A modal command on DM (KL mode #12)

Moreover, the modal basis approach provides better means to compare the performance of different AO designs.

Any wavefront, that is, the phase aberrations $\phi \in L^2(\Omega)$ can be written as an infinite set sum scaled of orthogonal basis functions (modes): $\phi(x,y) = \sum_{i=1}^{\infty} b_i f_i(x,y)$. The DM only has a finite number of actuators and can only take shapes defined by the DM influence functions (e.g., Gaussian influence functions, see Figure 3.4). Therefore, we consider a finite set of N modes in the DM space:

$$\phi(x,y) = \sum_{i=1}^{N} a_i m_i(x,y) + \phi_{res}, \tag{3.7}$$

where m_i are orthogonal DM space modes, a_i the modal coefficients and ϕ_{res} is part of the wavefront outside the finite set of DM modes.

The choice of the modal basis can be driven by the different system-dependent and user-dependent needs. However, a natural goal for the modal basis designing is to construct a modal basis that contains the maximum amount of turbulence energy for a given finite number of modes. That is, for a given N in Equation 3.7, we want to minimize ϕ_{res} in the least squares sense. Gendron and Léna (1994) showed that an optimal basis with this respect is obtained by considering the atmosphere's statistics and the DM influence function properties. The outcome of this process is called the Karhunen-Loève (KL) basis.

The construction method is based on double orthogonalization. We start by computing the ordinary KL decomposition of the atmosphere's spatial statistics. For Gaussian random fields such as the von Kàrmàn model, this is obtained with diagonalization of the covariance matrix of the discretized process. The KL decomposition yields the optimal basis in that it minimizes the total mean squared error (MSE). However, considering the DM influence functions, this basis is not orthogonal in the DM space. The problem is solved by projecting the modes on the influence functions of the mirror and re-orthogonalizing them (double orthogonalization). Details on the double diagonalization are found in Gendron (1994).

3.3 Adaptive optics control

Classically, an AO system is controlled by combining a linear reconstructor with a proportional-integral (PI) control law, often referred to as *the integrator*. As a starting point, it is assumed that the controller operates in a regime where the dependence between WFS measurements and DM commands is linear to a good approximation, satisfying

$$\boldsymbol{w}_t = Dv_t + \xi_t, \tag{3.8}$$

where $\boldsymbol{w}_t = (\delta w_t^1, \cdots, \delta w_t^n)$ is the WFS data, v_t the DM commands and D the so-called interaction matrix. Moreover, ξ_t models the measurement noise typically composed of photon and detector noise. The DM command vector v_t defines the DM shape given in the function subspace linearly spanned by the DM influence functions.

Calibration and the reconstruction matrix

The interaction matrix represents how the WFS sees each DM command (or the DM shape), and likewise, it presents a linear approximation of how the WFS sees phase error introduced by the atmosphere. More generally, it is a linear approximation of how the WFS sees any phase error introduced by the atmosphere or the DM (around small wavefront aberration). It can be derived mathematically if we accurately know the system components (WFS and DM) and the alignment of the system. But in practice, it is usually measured by probing the DM actuators or modal commands on DM (inside the linear range of the WFS) and recording the corresponding WFS measurements, as shown in Algorithm 1.

Algorithm 1 Calibration procedure

```
Initialize interactions matrix $D$
for actuators $act = 1$ in $1 \dots N^2$ do
    Set DM actuator act to $v$
    Record WFS measurement $\boldsymbol{w}_+$
    Set DM actuator act to $-v$
    Record WFS measurement $\boldsymbol{w}_-$
    Write the corresponding column in $D$ as $D[:, act] = (w_+ + w_-)/2v$
    set DM commands to zero
end for
```

The interaction matrix, obtained from the calibration, is generally ill-conditioned, and to invert it, a regularization method is needed; see Section 6.

The problem can be regularized by projecting v_t to a smaller dimensional subspace spanned by, for example, the KL modal basis. We denote the modal transformation matrix, which maps DM actuator voltages to modal coefficients by B_m. The modal interaction matrix (how the WFS sees the modes) is now obtained as $DB_m^\dagger$, where $B_m^\dagger$ is the Moore–Penrose pseudo-inverse of B_m. A well-posed reconstruction matrix for the inverse problem in (3.8) is then given by

$$C_m = (DP_m)^\dagger, \tag{3.9}$$

where $P_m = B_m^\dagger B_m$ is a projection map to the KL basis. Regularization by projection is a classical regularization with well-established theory (Engl et al., 1996). It is well-suited to the problem at hand due to the physics-motivated basis expansion and the fixed finite dimension of the observational data.

The integrator

Let us now give the full control law for the closed-loop system. In the following, we include Δ in the variable notations to highlight the residual variables in the closed-loop control, i.e., we write, e.g., $\Delta\boldsymbol{w}_t$ for the residual WFS data.

Namely, at a given timestep t, the WFS observes the residual wavefront, and the new control voltages $\tilde{\boldsymbol{v}}^t$ are obtained from

$$\tilde{\boldsymbol{v}}^t = \tilde{\boldsymbol{v}}^{t-1} + gC_m\Delta\boldsymbol{w}^t, \tag{3.10}$$

where g is so-called *integrator gain*. In literature, $g < 0.5$ is typically found to provide stable control for a two-step delay system (Madec, 1999). The integrator controller is constantly trying to drive down wavefront errors seen by the WFS toward zero. In the case of static phase error, it would eventually converge to a zero WFS measurement solution. However, it is a non-predictive control law; hence it is always lagging behind (due to the time delay) the evolving atmospheric turbulence.

3.4 Coronagraph

Even with optimally working extreme adaptive optics, the faint planet would often be lost within the much brighter diffraction rings of the host star; see Figure 3.5. The remedy for this is a coronagraph that suppresses the light from the brighter object. First introduced for observing hot gas surrounding the Sun by Lyot (1939), the fundamental principle of the coronagraph in HCI is the same – the coronagraph suppresses light from an on-axis source (e.g., the Sun or the host star) while preserving the off-axis companions' signal (e.g., the gas surrounding the sun or an exoplanet).

Nowadays, coronagraphy is an active research field in astronomical instrumentation, and the concepts have matured well beyond Lyot's basic design (Mawet et al., 2012). However, this thesis discusses XAO control and its effect on post-coronagraphic contrast. To this end, in numerical simulations, we use a theoretical ideal coronagraph model, where the coronagraphs suppress all light for an on-axis flat wavefront while preserving the off-axis source (Cavarroc et al., 2006). With the ideal model, the complex wavefront

Table 3.1: Simulations parameters for error term demonstration in Figure 3.6

Telescope		
Parameter	Value	Units
Telescope diameter	8	m
Obstruction ratio	14	percent
Sampling frequency	1000	Hz
Active actuators	1364	actuators
PWFS subapertures	41×41	apertures
PWFS modulation	0	λ / D
WFS wavelength	0.85	µm
Science camera wavelength	1.65	µm
NGS magnitude	0 & 10	low - & high noise
Integrator gain	0.6	
Atmosphere		
Fried parameter	16	cm @ 500 nm
Number of layers	3	$\cdots$
Layer altitudes	0 / 4 / 10	km
C_N^2	50 / 35 / 15	percent (%)
Wind speeds	10 / 26 / 35	m/s
Wind directions	0 / 45 / 180	degrees

ψ after the coronagraph on the pupil plane is given by

$$\psi_0(x, y) = \chi_\Omega(x, y) \left(\sqrt{E_c} - \exp[-i\phi(x, y)] \right), \tag{3.11}$$

where χ_Ω is the aperture and $E_c = \exp[-\sigma_\phi^2]$ the instantaneous coherent energy, and where σ_ϕ^2 is the spatial variance of the AO residual phase. The following focal plane PSF is calculated with the Fraunhofer approximation (2.4).

3.5 Estimating performance

The AO system's performance can be evaluated in several ways, of which the SR is probably the most common. As mentioned earlier, the SR is connected to residual wavefront variance via the Marechal approximations. In the case of integrator law, we can split the residual wavefront variance into four independent components:

$$\sigma_\phi^2 = \sigma_{fit}^2 + \sigma_{rec}^2 + \sigma_{temp}^2, \tag{3.12}$$

where σ_{fit} is the standard deviation of the fitting error, σ_{rec} the reconstruction error (containing the measurement noise, calibration noise, sampling errors, aliasing, chromaticity, and the errors due to the reconstruction method itself) and σ_{temp} the temporal error controlled by the systems time delay and loop frequency. While the DM properties set the fitting error, advanced control and reconstruction methods can mitigate the latter two error

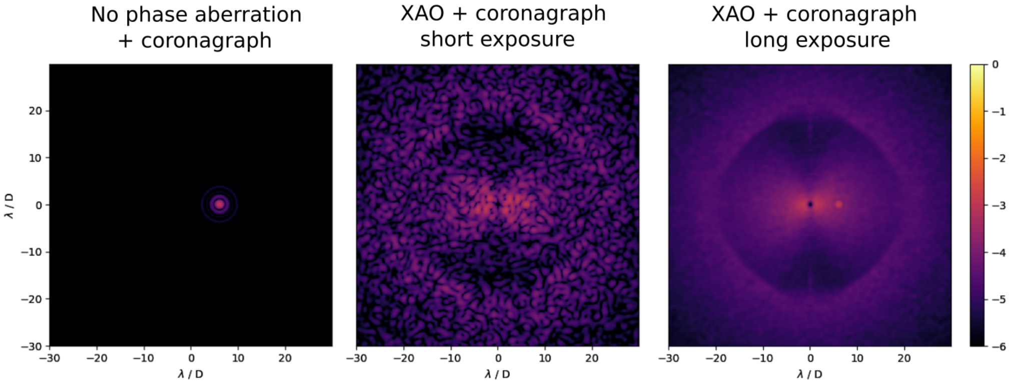


Figure 3.5: Normalised log-scale post-coronograph images showing image contrast. Coronagraph suppresses the light from the on-axis source while preserving the fainter off-axis source (located at $6\lambda/D$). With no phase aberrations, the ideal coronagraph suppresses all the light from the on-axis source. When we add the atmospheric turbulence and XAO control, the residual phase errors produce speckles in the image plane. On long exposure, these atmospheric speckles change over time and leave a smooth halo of stellar light in the image. The stellar light (speckles) in the post-coronagraph image limits sensitivity to observing faint exoplanets.

terms.

With a relatively bright guide star (magnitude < 8), the fitting error dominates the total phase variance and, consequently, the SR. On the other hand, with a faint star or challenging imaging conditions, the reconstruction error (namely the measurement noise) star is the dominating term. Typically, the temporal error has only a small impact on the SR. However, the ultimate goal of HCI is not only to deliver high SR but also to separate light from the exoplanet and the host star optically. To measure this ability, we use so-called *raw point spread function contrast*. It is defined as the ratio between post-coronagraphic PSF on the focal plane and a non-coronagraphic PSF peak intensity, that is,

$$s^0_\phi(u,v) = \frac{|\mathcal{F}\{\psi_0\}(u,v)|^2}{s_\phi(0,0)}, \tag{3.13}$$

where ψ_{coro} is the post-coronagraphic wavefront (3.11) and s_ϕ the PSF without coronograph. The raw PSF contrast gives the relative intensity of the starlight that leaked through the coronagraph to the image plane.

The wavefront errors are linked to the contrast by the superposition approximation (Guyon, 2005, 2018). Minimizing the residual wavefront maximizes the raw PSF contrast. Figure 3.6 illustrates how different error terms appear in the raw PSF contrast. All three main residual wavefront error terms show up in the raw PSF contrast as different features:

- *The wind-drive halo* (WDH): Connected to temporal error, this halo appears as a butterfly-shaped pattern in the focal plane image, spearing along the main directions of the wind; see the second row in Figure 3.6. It is most apparent when the wind is strong and the AO loop cannot keep up with the temporal changes leaving a residual phase with a clear directional pattern. When present, the WDH significantly reduces the obtained contrast, especially at small angular separations from their host star.

- *The control radius*: The finite number of DM actuators can only correct a finite set of spatial frequencies, from low-order frequencies up to so-called cut-off frequency. Low-order phase aberrations create speckles at small angular separations, while high-order errors create speckles further from the center. The first row in Figure 3.6 shows the raw PSF contrast of optimal XAO control; the only error is the fitting error. In the focal plane, the fitting error creates a central square-shaped dark zone defining the control radius. The speckles outside this area are due to the high-order aberrations that the DM cannot correct.

- *WFS measurement noise*: On fainter guide stars, the WFS collects fewer photons and, hence, the measurement contains more noise. The noise propagates into the wavefront reconstruction and induces speckles inside the whole control domain set by the control radius.

- *The aliasing error*: The WFS has a limited spatial sampling (e.g., the number of lenslets in SHS and the number of pixels in the detector for PWFS). Consequently, the atmosphere's high-order features get projected to low-order modes in the reconstruction. The efficiency of this aliasing process depends on the WFS type. For example, the SHS shows much stronger aliasing the PWFS (e.g., Vérinaud, 2004). The aliasing error can be mitigated with a spatial filter (Poyneer and Macintosh, 2004) or oversampling by the WFS, but a tiny amount of aliasing will already appear as features in the raw PSF contrast. Aliasing error is more substantial in the direction of the spatial grid of DM actuators; hence it shows as a cross-like pattern in the raw PSF contrast. Further, as high-order errors propagate slightly to the lowest-order modes, such as tip and tilt, the focal plane PSF is not well centered behind the coronagraphic mask, and starlight leaks through the coronagraph. The low-order residuals cause a diffraction-like pattern close to the center of the raw PSF contrast.

In addition to these error terms visible in numerical simulations, a real XAO system suffers from other error terms, for example, chromatic errors and temporal vibration, non-common path aberrations, as well as amplitude variations introduced by the atmosphere (scintillation). Chromatic errors appear because the effect of the refractive index on the atmosphere is not perfectly achromatic, and the WFS and science camera use different wavelengths. In small wavefront errors of XAO, the chromatic errors on science wavelength start to show up (Guyon, 2005). Further, in addition to atmospheric residuals, low-order residuals may arise from telescope vibration and differential thermo-mechanical effects. The low-order residuals, notably tip-tilt errors, push the PSF core away from the center behind the coronagraph focal plane mask.

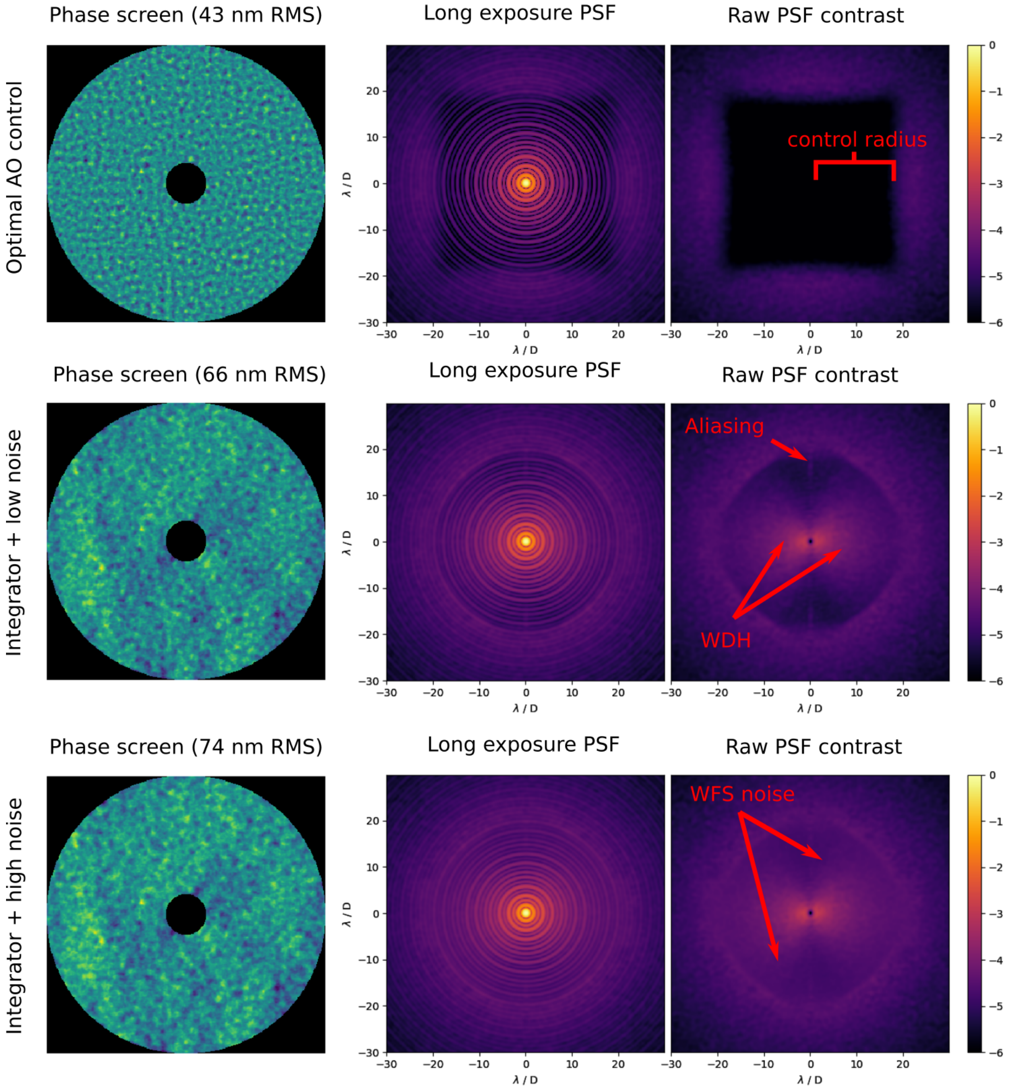


Figure 3.6: The effect of AO control and its error terms on raw PSF contrast. The raw PSF was calculated in three different cases: one with optimal AO control, i.e., only the fitting error present, and the other two with a PWFS and the integrator with two different noise levels; see parameter Table 3.1. Advanced control methods have the potential to push the raw PSF contrast in the second and third rows closer to the one obtained with the optimal XAO control (first row).

4 Machine learning

In recent years, there has been a growing focus on Machine Learning (ML) methods in the field of AO. ML is a research field that focuses on developing and understanding methods that are capable of ”learning,” which means they can leverage data to improve their performance on various tasks. Probably the most used and simplest to understand sub-category of machine learning problems is so-called supervised learning (SL).

SL is a type of machine learning that deals with problems where the data is labeled, meaning each data point contains features and an associated label (i.e., the output). The main objective of SL algorithms is to learn a function that maps input feature vectors to corresponding output labels based on example input-output pairs. The supervised learning algorithm infers the function from the training data, which consists of labeled examples. The algorithm then uses this inferred function to predict output labels for new input examples. The optimal scenario is for the algorithm to correctly predict the labels/outputs for unseen examples, requiring it to generalize from the training data to new, unseen situations.

4.1 Regression task

SL problems in AO control usually fall under the subcategory of regression tasks, where the inputs and outputs are continuously valued vectors. In a regression problem, the input data consists of a set of features or predictors $\mathbf{x} = (x_1, x_2, ..., x_n)$ and their corresponding continuous output values $\mathbf{y} = (y_1, y_2, ..., y_n)$. The goal is to learn a function $f_\theta(\mathbf{x})$, parameterized by a set of parameters θ, that can accurately predict $\mathbf{y}$ for new input data, by utilizing a labeled data set. More precisely, give an indexed data set $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^N$, we try to find a set of parameters θ that minimize a loss function $\mathcal{L}$ between the predicted output and label output. We have

$$\hat{\theta} = \arg\min_\theta \sum_{i=1}^N \mathcal{L}(f_\theta(\mathbf{x}^i), \mathbf{y}^i). \tag{4.1}$$

In the case of a common MSE loss function, it takes the form:

$$\hat{\theta} = \arg\min_\theta \sum_{i=1}^N \frac{||f_\theta(\mathbf{x}^i) - \mathbf{y}^i||^2}{N}. \tag{4.2}$$

To ensure that the learned model f_θ does not simply memorize the data but instead captures the underlying concepts, it is important to avoid *overfitting*. To evaluate the model's ability to generalize to new data, the available data is typically divided into two parts: the training data used to minimize the loss and find optimal parameters θ, and a separate set of data used to test the model's performance on unseen samples. A model is considered to be overfitted if a model performs well on the training data but poorly on the test data. Here's a general overview of the steps involved in training a regression model:

- *Splitting the data*: The next step is to split the data into training and validation sets.

The training set is used to fit the model, while the validation set is used to evaluate its performance and tune its hyperparameters.

- *Training the model*: The model is trained, that is, the model parameters θ are optimized, using an optimization algorithm that minimizes a loss function, such as mean squared error (MSE), on the training data. This involves updating the model parameters such that they minimize the loss function.

- *Model evaluation*: Once the model is trained, it is evaluated on the validation set to determine its performance.

- *Hyperparameter tuning*: The model's performance can be improved by tuning its hyperparameters, such as the learning rate, regularization strength, and the number of hidden layers. This involves searching over a range of hyperparameters and selecting the combination that produces the best performance.

- *Model deployment*: Once the model has been trained and its hyperparameters have been tuned, it can be deployed to predict new input data.

There are many different types of models, including linear regression, polynomial regression, neural networks, decision tree regression, and random forest regression, among others. Training a regression model involves fitting the model parameters to the training data to accurately predict the output values for new input data.

4.2 Neural networks

Artificial neural networks (ANNs), or shortly just neural networks (NN), are machine learning models loosely based on the structure and function of the human brain. NNs have become one of the most popular machine-learning techniques due to their ability to solve complex problems by learning from data. This section briefly describes the NN models used in this thesis, namely fully convolutional NNs. The interested reader can refer to Goodfellow et al. (2016) for details on NNs and their optimization.

Let us start by defining a fully connected NN, also called a multilayer perceptron (MLP). MLP is a feedforward NN, meaning the data flows in one direction from the input layer through the hidden layers to the output layer. Each NN layer consists of an affine transformation defined by the layer-wise weight matrix $\mathbf{W_i}$, bias $\mathbf{b}$, and a non-linear element-wise applied activation function z. The full model of l layer MLP is given by:

$$\mathbf{y} = \mathbf{W}_l z(\mathbf{W}_{l-1} \cdots z(\mathbf{W}_2 z(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \cdots) + \mathbf{b}_l, \tag{4.3}$$

where $\mathbf{x}$ is the input vector, $\mathbf{y}$ is the output vector. Common choices for the activation function are Tanh, sigmoid, and the rectified linear unit (ReLU) defined by $z(t) = \max(0, t)$. Another typical choice of activation function in image-to-image applications, such as AO control, is so-called *leaky rectified linear unit* (LeakyReLU) (Maas et al., 2013), defined by,

$$z(t) = \max(s * t, t), \tag{4.4}$$

where $s \in (0, 1)$ defines the slope of the negative input values.

The parameters to be optimized in MLP are the weights and biases, that is, $\theta = (\mathbf{W}_1, \mathbf{W}_2, \cdots \mathbf{W}_l, \mathbf{b}_1, \mathbf{b}_2, \cdots \mathbf{b}_l)$. The related model fitting problem is to find good or optimal parameters, θ in Equation 4.3 given some data set. The objective function in equation 4.1 does not have a closed-form solution for NNs; hence the parameters are optimized with some form of gradient descent algorithm, starting with random initialization of the parameters. This method involves computing the gradient of the cost function with respect to the weights and biases and updating them iteratively in the negative direction of the gradient. Give some initial value for the parameters $\theta_i \in \mathbf{R}^k$ (usually sampled from Gaussian distribution), where k is the total number of free parameters in the NN, the basic gradient descent step update for θ is given by

$$\theta_{m+1} = \theta_m - a\frac{\sum_{i=1}^{N} \delta\mathcal{L}(f_\theta i(\mathbf{x}^i), \mathbf{y}^i)}{\delta\theta}, \tag{4.5}$$

where a is the so-called learning rate, and N is the size of the data set or the size of a mini-batch sampled from the data set. There are several variants of gradient descent, including batch gradient descent, and stochastic gradient descent, each of which updates the parameters differently based on the size of the training data set by, for example, modifying a dynamic learning rate. The gradients of the neural network (NN) parameters are typically calculated using the back-propagation algorithm (Bishop and Nasrabadi, 2006), which utilizes the ordinary chain rule. Nowadays, the gradients are usually computed by using automated differentiation tools, such as PyTorch (Paszke et al., 2019) and TensorFlow (Dillon et al., 2017).

4.3 Convolutional neural networks

Convolutional neural networks (CNNs) are a type of NNs often used in, for example, computer vision tasks, such as image classification, object detection and segmentation (see, e.g., O'Shea and Nash, 2015). They are designed to process data with a grid-like topology, such as images, by applying convolutional filters to extract local features.

A CNN layer typically consists of multiple convolutional layers, sometimes followed by one or more fully connected layers. A CNN without a fully connected layer is called a fully convolutional NN. The convolutional layers contain a set of learnable filters, also known as kernels or weights, which are convolved with the input data to produce a feature map. Each filter extracts a specific feature from the input data, such as edges or corners, and the resulting feature maps are then passed to the next layer.

One advantage of CNNs is their ability to learn hierarchical representations of the input data. The lower layers of the network learn low-level features, such as edges and textures, while the higher layers learn more complex features, such as object parts and shapes. This hierarchical representation allows CNNs to perform better than traditional computer vision algorithms in tasks such as image recognition. Another key feature of CNNs is their parameter sharing, where the same set of weights is used to convolve the input data at different locations. This greatly reduces the parameters required to train the network, making it more efficient and less prone to overfitting. Further, in particular, for AO control, parameter sharing makes CNNs a powerful tool for learning data with homogeneous structures and recognizing typical reconstruction errors due to, for example, misregistration or optical gains.

More precisely, a single convolutional layer involves sliding the filters over the input data and computing the dot product between the filter weights and the corresponding input values at each location (discrete convolution). This operation produces a single output value stored in the output feature map. The filters are typically small in size, such as 3×3 or 5×5, and are applied at each location in the input data to produce a corresponding output value. The result of the convolution operation is a feature map that captures the presence of specific features in the input data at various spatial locations. The size of the feature map depends on the size of the input data, the size of the filters, and the stride of the convolutional operation. For standard two-dimensional CNNs, the input vector $\mathbf{x}_{i,j,l}$ is a stack of two-dimensional images. Stack contains also referred to as features, for example, the RGB channels of a color image for image recognition or the past time series of phase maps for AO control. A single convolutional layer is defined by

$$\boldsymbol{h}_{i,j,k} = z\left(\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{l=1}^{L}\boldsymbol{K}_{m,n,l,k}\mathbf{x}_{i+m-1,j+n-1,l}\right) \tag{4.6}$$

where $\mathbf{x}$ is the input tensor, $\boldsymbol{K}$ is the tensor of learnable convolutional filters, $\boldsymbol{h}$ is the output tensor, z is the activation function, M and N are the height and width of the filters, L is the depth of the input tensor (i.e., the length of image stack), and k is the index of the output channel (defining the number convolution of filters). The output tensor $\mathbf{h}$ has dimensions (H, W, K), where H and W are the height and width of the output feature map, respectively. The vector $\mathbf{h}$ is then passed to the consecutive layer. In addition, CNNs can contain skip-connections that connect non-consecutive layers together and pooling layers that down or up-scale the layer outputs (see, e.g.,O'Shea and Nash, 2015).

4.4 Uncertainty-aware neural network

This section introduces Uncertainty-aware NNs. Using machine learning to capture a non-deterministic system produces two types of uncertainty. The first one is called aleatoric uncertainty, which arises due to the stochastic nature of a system, such as observation and process noise. This type of uncertainty can be accounted for by mapping to a parameterization of a probability distribution while still training the network in a discriminative manner.

The second type of uncertainty is called epistemic uncertainty, which arises due to the lack of sufficient data to determine the underlying process exactly. In the limit of infinite data, this type of uncertainty should vanish. However, for data sets of finite size, it remains when predicting outputs from inputs. This thesis uses *probabilistic networks* to capture aleatoric uncertainty and *ensembles* to capture epistemic uncertainty.

Probablistic neural networks

In this thesis, we consider probabilistic NNs whose output neurons serve as parameters for a probability distribution function. It is important to note that this concept should not be mistaken for Bayesian inference - we do not set a prior distribution to NN parameters. This type of probabilistic NN can capture the aleatoric uncertainty. The probabilistic nature of NN does not restrict architecture; that is, they can be, for example, fully convolutional or connected NNs.

Using any tractable probabilistic distribution class, in general, is possible. However, assuming that the stochastic function we are trying to learn has a continuous output and a unimodal probability distribution, approximation by a Gaussian distribution is a common choice. We can represent the parameters of this distribution as nonlinear, parametric functions of the input data; that is, the probabilistic NNs is a neural network that takes data vector $\boldsymbol{x}$ as input and outputs a mean and variance for each output dimension in y, defining a conditional multivariate Gaussian distribution with a diagonal covariance matrix. That is,

$$\hat{p}_\omega(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mu_\omega(\mathbf{x}), \sigma^2_\omega(\mathbf{x})). \tag{4.7}$$

Instead of minimizing the square distance between the prediction and the label, the model is fitted by maximizing the log-likelihood of the parameters:

$$\hat{\theta} = \arg\max_{\omega} \quad \log\left(\prod_{i=1}^{N} \hat{p}_\omega(\mathbf{y}^i|\mathbf{x}^i, \mathbf{x}^i)\right). \tag{4.8}$$

The output of a network trained according to (4.8) models aleatoric uncertainty, where the output distribution is a function of the input. However, it cannot model epistemic uncertainty, which cannot be captured through solely discriminative training.

Ensembles

The common strategy to account for epistemic uncertainty is Bayesian inference, for example, and Gaussian process models capture epistemic uncertainty by default (Williams and Rasmussen, 2006). The Bayesian approach can also be applied to NNs (Neal, 2012). However, due to factors such as training speed, ease of implementation, minimal need for external parameter tuning, and acceptable performance, ensembles of bootstrapped models have become increasingly popular, especially in the field of reinforcement learning (Efron and Tibshirani, 1994; Chua et al., 2018; Osband, 2016). The concept of an ensemble of bootstrapped models is simple. Instead of using a single NN, we use a collection (ensemble) of NN with identical architecture. Each model has its unique data set to be trained upon that is bootstrap sampled (sampling with replacement) from the whole data. In practice, each model sees a different subset of data, leading to different NN approximations (especially in regions where data is sparse). We note here that the ensembles and epistemic uncertainty are not limited to probabilistic NNs - ensembles can also be composed of deterministic models.

In the case of probabilistic Gaussian NNs, the full ensemble defines a multimodal distribution as the output. In the case of deterministic models, the output is simply a discrete set of point estimates.

5 Reinforcement Learning

This section introduces the concept of Reinforcement Learning (RL) and the notation used in the articles. The idea is to introduce the central concept behind the methods used, and for more comprehensive material on RL, the reader may look, for example, at the book from Sutton and Barto (2018).

RL is an active branch of machine learning that learns a control task via interaction with the environment. The principal idea is to let the learner, called the *agent*, feed actions to the environment, observe the outcomes, and then improve the control strategy concerning some predefined reward signal. RL learning usually considers problems that have three essential characteristics: i) they are closed-loop problems where agents' actions influence their later inputs; ii) the agent does not have direct instructions as to what actions to take but learns which actions yield the biggest reward via trial and error; iii) the consequences, including the reward, of a specific action, play out over some future time horizon.

RL algorithms can be roughly divided into two categories, although the line between the algorithms can be somewhat blurred: the model-based reinforcement learning (MBRL) approach and the model-free reinforcement learning approach. The model-based approach uses a predictive environment model called the *dynamics model* (usually learned from the interaction of the agent and the environment) to answer the question: *what would happen if action $\boldsymbol{a}_t$ were applied at a given state?* The predictive model is then used to determine the sequence of the following "best possible" actions. The model-free methods avoid the environment's modeling by learning a control policy directly from the reward signals. Model-based methods are known to be sample-efficient compared to model-free methods (e.g., Atkeson and Santamaria, 1997; Kocijan et al., 2004; Deisenroth et al., 2013). A model-free algorithm such as policy gradient methods may need several orders of magnitude more iterations compared to model-based methods (Janner et al., 2019; Chua et al., 2018). On the other hand, since the model-free methods avoid the dynamics modeling step, they are unaffected by the restrictions in the predictive model, providing better performance in some control problems. This thesis focuses on the model-based approach.

5.1 Markov Decision Process

The de facto mathematical framework for the sequential decision problems in RL is the *Markov Decision Process* (MDP, Bellman, 1957), an extension of Markov chains. It is formulated as a 4-tuple $(\mathcal{S}, \mathcal{A}, p, \mathcal{R}_t)$, where $\mathcal{S}$ is a set of all possible states, $\mathcal{A}$ a set of possible actions, p state transition dynamics and R_t the reward function.

At time step t, MDP is in a state $\boldsymbol{s}_t \in \mathcal{S}$ and an agent then takes an *action* $\boldsymbol{a}_t \in \mathcal{A}$ based on the current state, and the *environment* changes to the next state $\boldsymbol{s}_{t+1}$ with respect to the transition dynamics $p : (\boldsymbol{a}_t, \boldsymbol{s}_t) \mapsto \boldsymbol{s}_{t+1}$, represented by a conditional probability density function $\mathcal{P}_t = p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ [2]. The transition dynamic only depends on the last state; hence they form a Markov sequence. At each time step, a *reward* $R_t = r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is also observed, which is a (possibly stochastic) function of the current state and action. The

[2] The initial state is drawn from the initial state distribution $\boldsymbol{s}_0 \sim p_0(\boldsymbol{s}_0)$

modeler usually designs the reward to make the agent produce some favorable behavior (e.g., correcting for turbulence distortions).

The standard MDP formulation assumes that the agent observes all relevant information on the state of the environment s_t. However, this is not the case in many real-world domains, such as adaptive optics control, where the state is observed through a partial or indirect observation $\boldsymbol{o}_t$. The so-called *Partially Observed Markov Decision Process* (POMDP) is a refined formulation for these problems. In POMDP, the agent has to decide the action based on the observations instead of the complete state. The underlying process, governed by the dynamics $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$, is still Markovian, but the observations $\boldsymbol{o}_t$ do not have to be. Consequently, past observations can still contain helpful information for the decision of the following action.

5.2 Model-Based Reinforcement Learning

This section details the MBRL framework and the notion. The essential element of model-based RL is to learn the approximate dynamics model for the true transition dynamics. The approximate model predicts how the system would behave if a specific sequence of actions were applied. The model is then used either to control the system with a *planning* algorithm or to train so-called *policy*, $\pi_\theta : \boldsymbol{s}_t \mapsto \boldsymbol{a}_t$, which is a parameterized function that maps states into actions.

This thesis considers dynamics that are parameterized as artificial NNs. The model can either be probabilistic NN or deterministic NN. More precisely, we approximate the conditional distribution $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ with a member of the parameterized distribution family $\hat{p}_\omega(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ satisfying

$$\hat{p}_\omega(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \sim \mathcal{N}(\mu_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t), \sigma^2_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)), \tag{5.1}$$

where the mean $\mu_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and the variance $\sigma^2_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$ of the Gaussian field are outputs of a neural network, and θ the NN weights and biases. For the deterministic approximation, we represent the dynamics simply with a parameterized function $\hat{f}_\omega(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$, outputting the vector representing the next state. Again parameters ω are the NN weights and biases. The deterministic approximate can be considered as a point estimator (e.g., conditional mean) of the true probabilistic dynamics.

Training the dynamics model

In MBRL, the dynamics model is learned from data collected by controlling the system itself and recording the outcome, that is, from interaction with the system. More precisely, the data is collected by sampling trajectories $\tau = (\boldsymbol{s}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{a}_{T-1}, \boldsymbol{s}_T)$, that is, controlling the system for a certain number of timesteps and recording the sequence of states (observations for POMDP) and actions applied. These chunks of data are called *episodes*. The first few trajectories τ are collected by executing random actions, while the following trajectories are collected by controlling with a planning algorithm or policy. After episodes the trajectories τ are then sliced into timestep-wise training data inputs $(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and corresponding output label $\boldsymbol{s}_{t+1}$. The data is collected into a data set $\mathcal{D}$.

The probabilistic dynamics model is trained by maximizing the log-likelihood of a Gaussian for which the parameters are outputs of the neural network model. More specifically, given a dataset of N transitions $\mathcal{D} = \{(\boldsymbol{s}_t^i, \boldsymbol{a}_t^i), \boldsymbol{s}_{t+1}^i\}_{i=1}^N$ we maximize the following objective function

$$\hat{\theta} = \arg\max_{\omega} \quad \log\left(\prod_{i=1}^{N} \hat{p}_\omega(\boldsymbol{s}_{t+1}^i | \boldsymbol{s}_t^i, \boldsymbol{a}_t^i)\right) \tag{5.2}$$

where $\hat{p}_\omega$ is given by equation (5.1). The deterministic model is trained simply by minimizing MSE between the prediction and data set labels:

$$\frac{1}{|N|}\sum_{\mathcal{D}} \left\| \boldsymbol{s}_{t+1} - \hat{f}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t) \right\|^2 = \frac{1}{|N|}\sum_{\mathcal{D}} \left\| \boldsymbol{s}_{t+1} - \hat{\boldsymbol{s}}_{t+1} \right\|^2, \tag{5.3}$$

where $\boldsymbol{o}_{t+1}$ is obtained from the state $\boldsymbol{s}_{t+1}$ and $\hat{\boldsymbol{s}}_{t+1}$ is the observation predicted by $\hat{f}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$. Both optimization problems, (5.2) and (5.3), can be solved via backpropagation with stochastic gradient decent algorithms, such as the Adam algorithm. The Adam optimizer is a stochastic gradient descent algorithm that utilizes an adaptive learning rate, and it has shown great performance in many deep learning applications in computer vision, natural language processing, and RL (Kingma and Ba, 2014).

It is well-known that model-based RL, with NN models, unfavorably exploits an overfitted dynamics model in control, especially in the early stages of training (Nagabandi et al., 2018). This arises from the fact that ordinary NNs cannot capture epistemic uncertainty. To discourage this behavior, we employ an ensemble of several models (deterministic or probabilistic), each of which is trained using different bootstrap datasets, that is, subsets of the observations collected during training. Ensemble of models is a coarse approximation of epistemic uncertainty but still provides reasonable, effective uncertainty estimates for MBRL; see Chua et al. (2018) for a more detailed discussion. In the following sections, we denote both the deterministic and probabilistic dynamics by $\hat{p}_\omega$.

Model Predictive Control

Once we have collected the first trajectories of data and trained a reasonable approximation of the system dynamics, we may use the learned dynamics model to plan for a sequence of actions to be applied. The goal of the planning algorithm is to optimize a sequence of actions $\{a_t, a_{t+1} \cdots a_{t+T}\}$ such that it maximizes the expected reward inside some planning horizon T (Camacho and Alba, 2013). In planning, the dynamics model is used in a recursive manner – a Markov state will evolve from one timestep to the next with respect to the approximate predictive model, e.g., $\boldsymbol{s}_{t+2} \sim \hat{p}_\omega(\boldsymbol{s}_{t+2}|\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1})$, where $\boldsymbol{s}_{t+1} \sim p_\omega(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. The planning task can be formulated as an optimization problem:

$$(\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t+H}) = \arg\max_{\boldsymbol{a}_{t:t+H}} \mathbb{E}_{\hat{p}_\omega}\left[\sum_{h=0}^{H} r(\tilde{\boldsymbol{s}}_{t+h}, \boldsymbol{a}_{t+h})\right], \tag{5.4}$$

where

$$\tilde{\mathrm{s}}_1 = \mathrm{s} \quad \text{and} \quad \tilde{\mathrm{s}}_{t+1} = \hat{p}_\omega(\tilde{\boldsymbol{s}}_t, \boldsymbol{a}_t).$$

and $(\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \dots, \boldsymbol{a}_{t+H}) = \boldsymbol{a}_{t:t+H}$. We note here that the state propagation (5.2) depends on the dynamics model and propagation method used. The next state can either be a distribution or a point estimate, and the details for different cases can be found in the articles. Instead of planning the optimized action sequence once, at every timestep, the controller only executes the first action, receives a new state/observation from the system, and then recalculates the optimal action sequence. This procedure of re-planning at each timestep is referred to as *model predictive control* (MPC). The pseudo-code of MBRL with MPC planning is given in Algorithm 2.

Algorithm 2 Model-based RL with MPC

```
Initialize dynamics model parameters $\omega$ randomly
Initialize gradient iteration length $K$, batch size $B < |\mathcal{D}|$ and planning horizon $H$
Generate samples $\{s_{t+1}, s_t, a_t\}$ by taking random actions for $T$ timesteps (an episode)
    and append to $\mathcal{D}$
while not converged do
    Fit dynamics by minimizing Eq. (5.3) w.r.t. $\omega$ using Adam
    for iteration $t = 1$ to $T$ do
        Observe current state $\boldsymbol{s}_t$
        Use the dynamics $\hat{p}_\omega$ to optimize the action sequence w.r.t. (5.4)
        Save the interaction data $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$
    end for
end while
```

Model-based Policy Optimization

The MPC algorithm of model-based RL is often iterative and could, therefore, be too slow for some applications, such as the XAO control. However, if the chosen reward function is differentiable, the dynamics model can be used to optimize a policy function π. In particular, we optimize a set of parameters of policy π_θ, where θ is the set of parameters of the policy, in our case, the weights and biases of a neural network. The control decision is then only a single forward pass of a neural network, given by $a_t = \pi_\theta(\boldsymbol{s}_t)$. In policy optimization, we wish to find the parameters θ that maximize the expected cumulative reward the agent receives, that is,

$$\arg\max_\theta \mathbb{E}_{p_\theta(\mathrm{s}_0,\dots,\mathrm{s}_T)} \left[\sum_{t=0}^{T} r(\boldsymbol{s}_t, \pi_\theta(\boldsymbol{s}_t)) \right], \tag{5.5}$$

where

$$p_\theta(\mathrm{s}_0, ..., \mathrm{s}_T) = p_0(\boldsymbol{s}_0) \prod_{t=1}^{T} p(\boldsymbol{s}_t | \mathrm{s}_{t-1}, \pi_\theta(\mathrm{s}_{t-1}))$$

with the initial distribution $\boldsymbol{s}_0 \sim p_0$ and convention $\pi_\theta(\mathrm{s}_{-1}) = \boldsymbol{a}_0$ for a fixed initial action $\boldsymbol{a}_0$. However, as we do not have access to the true dynamics model p, we must approximate it with the learned dynamics model $\hat{p}_\omega$. Since the full length of the experiment T might not be fixed or/and very long, like in MPC, the policy is optimized only on actions over an extended time horizon $H \ll T$. Let us define

$$\hat{r}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}(g(\tilde{\boldsymbol{s}}_{t+1})) \tag{5.6}$$

where $\tilde{\boldsymbol{s}}_{t+1} = \hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and $g(.)$ a function that evaluates the reward of next state. This leads to the approximate policy optimization problem

$$\arg\max_\theta \sum_{\mathrm{s} \in \mathcal{D}} \sum_{t=1}^{H} \hat{r}_\omega(\tilde{\boldsymbol{s}}_t, \pi_\theta(\tilde{\boldsymbol{s}}_t)), \tag{5.7}$$

where H the planning horizon and

$$\tilde{\mathrm{s}}_1 = \mathrm{s} \quad \text{and} \quad \tilde{\mathrm{s}}_{t+1} = \hat{p}_\omega(\tilde{\boldsymbol{s}}_t, \pi_\theta(\tilde{\boldsymbol{s}}_t)).$$

Again the state propagation (5.2) depends on the choice of dynamics model and the method itself, and the details are left for the articles themselves. Model-based optimization utilizes the differentiable nature of both our models and the reward function. The gradients of policy parameters ω can be calculated by back-propagating through rewards collected along the planning horizon. The optimization (5.7) is then carried out with stochastic gradient descent, such as the Adam algorithm. A pseudo-code for generic policy optimization is given in Algorithm 3.

Algorithm 3 Model-based Policy Optimization

```
Initialize policy and dynamics model parameters $\theta$ and $\omega$ randomly
Initialize gradient iteration length $K$, batch size $B < |\mathcal{D}|$ and planning horizon $H$
Generate samples $\{s_{t+1}, s_t, a_t\}$ by taking random actions for $T$ timesteps (an episode) and append to $\mathcal{D}$
while not converged do
    Fit dynamics by minimizing Eq. (5.3) w.r.t $\omega$
    Improve the Policy (i.e., optimize model parameters $\theta$) by minimizing Eq. (5.5) with update dynamics
    Generate samples $\{s_{t+1}, s_t, a_t\}$ by running policy $\pi_\theta(a_t|s_t)$ for $T$ timesteps (an episode) and append to $\mathcal{D}$
end while
```

6 Adaptive optics as an RL problem

Let us recall the main RL problem characteristics: i) closed-loop problem; ii) ”No instruction” on actions that maximize the reward; iii) delayed the consequences of actions. Keeping these characteristics in mind, in principle, RL for XAO provides a path to resolve control strategies that understand the dynamic range and sensitivity of the WFS, do not rely on accurate calibration of the system, and learn the temporal bandwidth and delay of the system. Further, not studied in the thesis, RL also provides a flexible framework for design controllers that utilize reward functions from multiple sensors (sensor fusion) to tackle, for example, chromatic or non-common path errors in the XAO loop.

The XAO control, by its nature, differs from standard ”benchmark” RL problems. The usual reward functions for XAO control are not sparse in the environment but distributed rather evenly through time, and the effective time horizon (the consequence of action) is relatively short. In the case of a simple two-frame delay, no DM dynamic, and no noise, we would plan to minimize the observed wavefront sensor measurements two steps into the future; that is, we would implicitly predict the best control action by the DM at the time of the corresponding WFS measurement. However, the effective planning horizon is longer in the presence of DM dynamics and temporal jitter since the control voltage decisions are not entirely independent. The choice of the planning horizon compromises two effects: too short a planning horizon jeopardizes the loop stability, and too long a planning horizon makes the method prone to overfitting. Planning horizon $H = 4$ is a reasonably well-working compromise experiment (Nousiainen et al., 2021).

The challenge of XAO control comes from 1) the vast control space (1000 – 10k actuators), 2) the cross-correlation of the actuators, 3) the indirect observation of the system, 4) extreme time constraint on control (from kilohertz to several kilohertz), and 5) the method’s ability to adapt to atmospheric conditions online. Therefore, RL algorithms for XAO need special consideration. We may approach the RL for XAO challenges by asking two related questions: ”how to model the XAO as an MDP?” and ”how to design an RL algorithm that archives the XAO requirements?”. The solutions for both of the questions determine the properties of the controller. In the following, we discuss the MDP formulation and XAO-specific solutions used in this thesis.

Adaptive optics as a Markov decision process

Before formulating AO as an MDP, let us first consider a discrete-time state-space model of an AO system with one WFS and DM; the science camera is not included in the observation model. The states consist of all the information needed to ensure the Markov property. Taylor’s frozen flow model combined with step-wise linear DM response yields the following Markov model. We denote the turbulence at each layer, that is, a collection of phase aberration after each layer, by vector $\Phi_t = [\phi_1, \phi_2, \ldots, \phi_L]$, where L is the number of turbulence layers, and the DM surface by φ^{DM}. The state of the system $\boldsymbol{x}_t$ is the DM shape and the turbulence Φ_t over the telescope, that is, $\boldsymbol{x}_t = [\Phi_t, \varphi_t^{DM}]^\top$. Neglecting the control delay, we may formulate AO control as follows,

$$\begin{aligned} \boldsymbol{x}_{t+1} &= \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} \boldsymbol{x}_t + \begin{bmatrix} 0 \\ F \end{bmatrix} \boldsymbol{v}_t + \epsilon_t \\ \boldsymbol{w}_{t+1} &= \Gamma\left([P, -I]\boldsymbol{x}_{t+1}\right) + \beta_{t+1}, \end{aligned} \tag{6.1}$$

where ϵ and β are Gaussian i.i.d. noise, T an operator that shifts each turbulence layer with respect to their wind speed and direction, F the projection of voltages to DM influence functions, P a projection of all turbulence layers along the line of sight and I the identity matrix. Further, the Γ represents the WFS measurement model (see Section 3.1). We note that the time evolution model operates with a full turbulence profile. The control delay can be added to this model by extending the state space with past commands $\boldsymbol{v}$. The most common AO simulators simulate XAO with this kind of evolution model. However, it is still a simplified approximation of the accurate AO system and atmosphere. Moreover, information required for such an evolution model (i.e., the C_N^2 profile and wind speeds and directions) is unknown a priori, and an accurate estimate of these is difficult to obtain.

Let us next consider an alternative state space model with MDP notation and formulation. The measurement model in (6.1) is indirect in two ways: first, WFS measures the cumulative phase aberration through the layers, and second, the measurement itself is indirect. Using this Markov model would require a full reconstruction of Φ. However, XAO control aims to apply DM commands that minimize the future (cumulative) phase errors observed in the sensor. In other words, the target state of the system is a state that gives a flat reference measurement (calibrated for optimal PSF). A natural goal, defined by the reward function, is the negative distance between the observation and the target state observation – the closer to flat reference, the more reward. Assuming that flat reference gives zero WFS measurement (no NCPA), the reward for an action $\boldsymbol{a}_t$ at a state $\boldsymbol{s}_t$ is given by

$$r(\boldsymbol{s}_t, \boldsymbol{a}_t) = -\mathbb{E}_p \|\boldsymbol{o}_{t+1}\|^2, \tag{6.2}$$

where $\boldsymbol{o}_t$ is the post-processed WFS measurement (either camera intensities, slopes, or projection to voltages), referring to the observation of MDP, and p is the true dynamics of the system. Using this reward function, we do not need to approximate the full state of the system but only how actions and atmosphere affect the following observations (WFS measurements). However, the observations from the system do not follow Markovian statistics – the future observation $\boldsymbol{o}_{t+1}$ is not only dependent on previous observation $\boldsymbol{o}_t$ and action $\boldsymbol{a}_t$. One solution to non-Markovian dynamics is to extend the state space with past observations and actions to guarantee approximately Markovian behavior. That is, we concatenated previous observations and actions to form the MDP's state such that

$$\boldsymbol{s}_t = \left(\boldsymbol{o}_t, \boldsymbol{o}_{t-1}, \ldots, \boldsymbol{o}_{t-k}, \boldsymbol{a}_{t-1}, \boldsymbol{a}_{t-2}, \ldots, \boldsymbol{a}_{t-m}\right), \tag{6.3}$$

The action of the MDP is simply the set of integral control voltages send to DM actuators

$$\boldsymbol{a}_t = \Delta\tilde{\boldsymbol{v}}_t. \tag{6.4}$$

With this formulation, we can replace the high dimensional state space model (6.1), by approximate dynamics $p(\boldsymbol{o}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ and bypass the reconstruction for complete phase profile Φ. However, the problem of indirect WFS remains.

Ill-posed observation model

Let us consider an indirect WFS measurement of the DM commands. As discussed in Section 3.1, on closed-loop residuals, the measurement model can be approximately modeled by a linear equation:

$$\boldsymbol{w} = D\boldsymbol{v} + \xi, \tag{6.5}$$

where ξ is a noise term and D the interaction matrix; see Section 3.3. Further, $\boldsymbol{v} \in \mathbb{R}^n$ and $\boldsymbol{w} \in \mathbb{R}^m$, where n is the number of DM actuators and $m/2$ the number of SHS lenslets/PWFS pixels (2 slope measurement for each subaperture). The interaction matrix models how WFS measurements $\boldsymbol{w}$ and control voltage (DM commands) $\boldsymbol{v}$ are related via physics. The field of inverse problems considers the inverse of the direct modeling problem (6.5). That is, to solve $\boldsymbol{v}$, given the related measurement $\boldsymbol{w}$. This task is non-trivial since the underlying mathematical model is ill-posed (Engl et al., 1996). By the classical definition of Hadamard, a problem is called ill-posed (as opposed to well-posed) when at least one of the following conditions is violated: i) a solution exists ii) the solution is unique, (iii) the solution has to depend continuously on the data.

The problems with existence i) and uniqueness conditions ii) can be dealt with using the Moore-Penrose pseudoinverse. However, the violation of the stability condition iii) typically leads to numerical challenges in inverse problems that for problem (6.5) appear as a high condition number of the matrix. Consequently, even if D were invertible (no pseudoinverse needed), a naive reconstruction by $D^{-1}\boldsymbol{w} = \boldsymbol{v} + D^{-1}\xi$, would lead to useless reconstruction since the size of the noise term would be potentially multiplied by the big conditioning number of the D. This phenomenon is called noise amplification and is also prominent for pseudoinverse. For reinforcement learning with ill-posed observation, noise amplification means that actions arbitrarily far from each other can yield very similar observations. The RL agent is ”blind” to some consequences of actions it takes. Consequently, if not dealt with, it jeopardizes the stability of RL.

A widely used solution to deal with noise amplification is to use so-called *truncated singular value decomposition* (TSVD). In TSVD, the interaction matrix D is first factorized with standard SVD, that is, $D = VSU^\top$, where V is a $m \times m$ orthogonal matrix, S is a $m \times n$ diagonal matrix containing $\min(n, m)$ singular values (if D is not full rank then rank(*D*) singular values) and U an $n \times n$ orthogonal matrix. Then, the inverse of the diagonal matrix is truncated so that the diagonal elements below some certain threshold are set to zero. Let us denote the truncated inverse of S by $S^\dagger_\alpha$, where the α is the truncation threshold. Now we can define a well-posed reconstruction function F_α by the formula

$$F_\alpha(\boldsymbol{w}) = US^\dagger_\alpha V^\top \boldsymbol{w}. \tag{6.6}$$

The corresponding reconstruction $\hat{\boldsymbol{v}} = F_\alpha(\boldsymbol{w})$ is a linear combination of the columns in V corresponding to the non-truncated singular values. The reconstructed solution

$\hat{\boldsymbol{v}} = F_\alpha(\boldsymbol{w})$ in the subspace is spanned by these column vectors, that is, the problem is regularized by projection to a smaller subspace (Mueller and Siltanen, 2012). We denote a collection of these column vectors by V_α. Similarly, for RL if the command vector $\boldsymbol{v}$ is projected to these column vectors (i.e., modes), the consequences (i.e., phase aberrations introduced by the DM) of the action are ”seen” in the measurement. This projection turns an ill-posed observation of the RL problem into a well-posed one. More precisely, for MBRL, we add a constraint to MPC optimization (5.4) as follows,

$$(\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t+H}) = \underset{\boldsymbol{a}_{t:t+H} \in \operatorname{span}\{V_\alpha\}}{\arg\max} \; \mathbb{E}_{\hat{p}_\omega} \left[\sum_{h=0}^{H} r(\tilde{\boldsymbol{s}}_{t+h}, \boldsymbol{a}_{t+h}) \right], \tag{6.7}$$

where V_α is a collection of singular vectors corresponding to non-truncated singular values. For model-based optimization, we restrict the policy model output to the subspace spanned by the singular vectors V_α. That is, we add a filter to the output layer of the generic NN policy in Equation (5.7):

$$\pi_\theta(\boldsymbol{s}_t) = P_\alpha G_\theta(\boldsymbol{s}_t), \tag{6.8}$$

where P_α is an orthogonal projection onto $\operatorname{span}\{V_\alpha\}$ and G_θ is a standard NN, where the output is vectorized.

The truncation parameter α determines the number of singular vectors (modes) used in the reconstruction. It balances the inversion between stability and accuracy, that is, the level of noise amplification and reconstruction details. The standard TSVD only considers the measurement modality – it does not encode any prior information on the reconstruction. However, in adaptive optics, we know that the atmospheric turbulence approximately follows von Kàrmàn statistics. By utilizing the KL modes introduced in Section 3.2, we can truncate the inversion matrix in such a way that it contains the required turbulence energy (reconstruction accuracy) with a minimal number of modes (noise amplification).

From the inverse problem perspective, one could consider alternative regularization strategies, such as the Tikhonov regularization (Engl et al., 1996) of the Bayesian approach (Kaipio and Somersalo, 2006). These could be implemented by adding a regularization term in the optimization problem (5.4). However, the TSVD of the KL basis has some advantages in the framework of XAO control. Firstly, the number of KL modes needed for stable reconstruction covers enough turbulence energy for XAO control. In other words, optimal reconstruction in the subspace spanned by these modes is adequate for XAO, and truncating the number of modes removes the part of the control space which is most insignificant for the control of atmospheric turbulence. Secondly, the basis computation (SVD) can only be done before the science operations, and any modern hardware on the telescope has the computational capacity to do it.

7 Discussion on results

7.1 Article I

This thesis aimed to develop data-driven control techniques that learn predictive and noise-robust control straight from the system feedback without prior knowledge of the system's modeling errors, such as the misregistration of DM and WFS. A natural framework for such algorithms is RL. This article was a starting point for using model-based RL algorithms for XAO control. Along with Landman et al. (2020, 2021), it was also the first paper that discussed the prospect of using RL for XAO to suppress the temporal error.

Our key result in the article is the formulation of the adaptive optics task as an MDP and a way to deal with an ill-posed observation. Further, using this formulation, we adapted a standard state-of-the-art MBRL algorithm, Probabilistic Ensemble Trajectory Sampling (PETS, Chua et al., 2018), to solve the AO task. We showed that the method suppresses temporal error and measurement noise and also learns to compensate for the misregistration between the WFS and the DM. The algorithm and MDP formulation presented was only one way to solve the control loop with RL but already hinted at the great potential of MBRL control for AO. The paper also discusses the limitations of the proposed method, especially the significant hurdle of inference time and computational jitter.

7.2 Article II

This article continues on the same topic. Notably, it focuses on the shortcomings of the first paper's method. Instead of running a computationally costly MPC algorithm at each time step, we utilize the dynamics model to train a policy NN to control the system. The policy NN scales to sub-millisecond inference for both VLT-scale and ELT-scale XAO systems. This refined method is called algorithm Policy Optimization for Adaptive Optics (PO4AO).

We introduced the PO4AO and studied its properties in extensive numerical simulation, confirming the predictive power, noise reduction, and its ability to compensate for the modeling errors of linear reconstruction, such as the optical gain effect of non-modulated PWFS. Further, we implemented PO4AO in a laboratory setup using the Magellan Adaptive Optics eXtreme system (MagAO-X, Males et al., 2018), and observed that the results were in line with the results from numeric simulations.

The work presents a significant step forward for XAO control with RL. The results indicate that RL is a promising approach for XAO control and can potentially solve many challenges in XAO control simultaneously. Further, the results show that it is possible to control the current and future XAO systems with existing hardware.

7.3 Article III

The PCS instrument for ELT will have a cascaded XAO design, meaning that the XAO system is placed after a regular AO system. The temporal dynamics of this second-stage system are also affected by the first-stage AO system. This paper discusses the prospect of running such a system with MBRL. For this, the algorithm was implemented on the GPU-based High-order adaptive OpticS (GHOST) bench at ESO headquarters, which simulates a second-stage AO system by running numerically simulated residual turbulence-phase screens across a programmable Spatial Light Modulator (SLM). Further, we introduced refinements to the original algorithm (PO4AO), derived tuned hyperparameters, analyzed corresponding results against a well-tuned integrator controller, and discussed future work briefly.

The paper demonstrates the method's ability to reproduce the promising results from stable numeric simulation in a laboratory setup with real PWFS and DM. The highly robust method performs better than a well-tuned integrator in challenging conditions (high wind speeds, faint NGS) with less predictable second-stage turbulence. A natural conclusion of the paper is that PO4AO is ready for on-sky testing.

7.4 Article IV

Another goal of the dissertation was to open discussion on RL methods for XAO. To this end, we prepared an open-source OpenAI Gym (Brockman et al., 2016) interface for several AO simulators called FitAO. FitAO is a concept platform designed to enable algorithmic development on multiple end-to-end AO simulation environments. Moreover, it is configured to utilize the interface specifications of the OpenAI Gym to allow the integration of modern control algorithms in the Gym library. We reviewed the functionality and design of FitAO and demonstrated its capabilities with a simple tutorial on applying reinforcement learning to the classical integrator control in a closed-loop SCAO system.

7.5 Conclusion

To summarize, RL offers a promising alternative for XAO control schemes. If well formulated, RL can simultaneously solve several challenges in XAO control, such as misregistration, photon noise, and temporal error. Additionally, RL is resilient to issues that arise when transitioning from simulations to real-world scenarios, such as data mismatch and non-Gaussian noise. The algorithms we developed in this thesis could be implemented in on-sky systems with existing hardware. Further, the methods discussed in this thesis are highly parallelizable and scale up to systems with 10^4 DoF, both the performance and inference time-wise. Once the RL method is implemented and tuned, it turns AO control into a turn-key operation - it adapts to changing conditions and dynamic misregistration.

Even though the main contribution of the thesis is in the field of astronomical instrumentation, the thesis results can also be positioned concerning inverse problems and RL research fields. For RL, AO control provides an interesting application for two reasons. Firstly, it is an example of a control problem with unusually high-dimensional action space but short planning horizon, giving variability to existing ”benchmark” RL problems. Secondly, and more importantly, as deep learning and RL methods are transforming many fields, such as protein folding, inverse problems, and robotics, there is potential for the same for direct exoplanet imaging. This thesis shows that RL can mitigate several decisive error terms that set limits for direct exoplanet imaging. Hence, when further developed and implemented in ELT-era telescopes, RL solutions may be instrumental in achieving the first direct images of habitable exoplanets – RL methods can play a significant role in obtaining revolutionary astronomical observations.

Further, the most classical practical applications of inverse problems, such as computational and electrical impedance tomography, are static and non-invasive, where the reconstruction does not affect the measurements. For the inverse problems research community, AO control is an interesting example of a control problem (RL problem) where the reconstruction (action) at a given state affects the subsequent measurement, and the system's state is observed through an ill-posed measurement. This enables more theoretical work on RL with ill-posed observations with a practical application.

References

Atkeson, C.G. and Santamaria, J.C. (1997). A comparison of direct and model-based reinforcement learning. In: *Proceedings of international conference on robotics and automation*, vol. 4, pp. 3557–3564. IEEE.

Babcock, H.W. (1953). The possibility of compensating astronomical seeing. *Publications of the Astronomical Society of the Pacific*, 65(386), pp. 229–236.

Basden, A.G., Myers, R.M., and Gendron, E. (2011). Wavefront sensing with a brightest pixel selection algorithm. *Monthly Notices of the Royal Astronomical Society*, 419(2), pp. 1628–1636. ISSN 0035-8711, doi:10.1111/j.1365-2966.2011.19825.x, url: `https://doi.org/10.1111/j.1365-2966.2011.19825.x`.

Bellman, R. (1957). A Markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684.

Beuzit, J.L., et al. (2010). Direct detection of giant extrasolar planets with SPHERE on the VLT. In: *Pathways Towards Habitable Planets*, vol. 430, p. 231.

Bishop, C.M. and Nasrabadi, N.M. (2006). *Pattern recognition and machine learning*, vol. 4, 4. Springer.

Brockman, G., et al. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Camacho, E.F. and Alba, C.B. (2013). *Model predictive control*. Springer Science & Business Media.

Cavarroc, C., et al. (2006). Fundamental limitations on Earth-like planet detection with extremely large telescopes. *Astronomy & Astrophysics*, 447(1), pp. 397–403.

Chua, K., Calandra, R., McAllister, R., and Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In: *Advances in Neural Information Processing Systems*, pp. 4754–4765.

Conan, J.M., et al. (2011). Are integral controllers adapted to the new era of ELT adaptive optics? In: *AO4ELT*.

Correia, C., Raynaud, H.F., Kulcsár, C., and Conan, J.M. (2010a). On the optimal reconstruction and control of adaptive optical systems with mirror dynamics. *JOSA A*, 27(2), pp. 333–349.

Correia, C., et al. (2010b). Adapting optimal LQG methods to ELT-sized AO systems. In: *1st AO4ELT conference-Adaptive Optics for Extremely Large Telescopes*, p. 07003. EDP Sciences.

Correia, C.M., et al. (2017). Modeling astronomical adaptive optics performance with temporally filtered Wiener reconstruction of slope data. *JOSA A*, 34(10), pp. 1877–1887.

Davies, R. and Kasper, M. (2012). Adaptive optics for astronomy. *Annual Review of Astronomy and Astrophysics*, 50, pp. 305–351.

Deisenroth, M.P., Fox, D., and Rasmussen, C.E. (2013). Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2), pp. 408–423.

Dessenne, C., Madec, P.Y., and Rousset, G. (1998). Optimization of a predictive controller for closed-loop adaptive optics. *Applied optics*, 37(21), pp. 4623–4633.

Dillon, J.V., et al. (2017). Tensorflow distributions. *arXiv preprint arXiv:1711.10604*.

Doelman, N. (2020). The minimum of the time-delay wavefront error in adaptive optics. *Monthly Notices of the Royal Astronomical Society*, 491(4), pp. 4719–4723.

Efron, B. and Tibshirani, R.J. (1994). *An introduction to the bootstrap*. CRC press.

Ellerbroek, B.L. and Vogel, C.R. (2009). Inverse problems in astronomical adaptive optics. *Inverse Problems*, 25(6), p. 063001.

Engl, H.W., Hanke, M., and Neubauer, A. (1996). *Regularization of inverse problems*, vol. 375. Springer Science & Business Media.

Fauvarque, O., et al. (2016). General formalism for Fourier-based wavefront sensing. *Optica*, 3(12), pp. 1440–1452.

Fauvarque, O., et al. (2017). General formalism for Fourier-based wave front sensing: application to the pyramid wave front sensors. *Journal of Astronomical Telescopes, Instruments, and Systems*, 3(1), p. 019001.

Ferreira, F., Gratadour, D., Sevin, A., and Doucet, N. (2018). COMPASS: an efficient GPU-based simulation software for adaptive optics systems. In: *2018 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 180–187. IEEE.

Fried, D.L. (1966). Optical resolution through a randomly inhomogeneous medium for very long and very short exposures. *JOSA*, 56(10), pp. 1372–1379.

Fusco, T., et al. (2006). High-order adaptive optics requirements for direct detection of extrasolar planets: Application to the SPHERE instrument. *Optics Express*, 14(17), pp. 7515–7534.

Gendron, E. (1994). Modal control optimization in an adaptive optics system. In: *European Southern Observatory Conference and Workshop Proceedings*, vol. 48, p. 187.

Gendron, E. and Léna, P. (1994). Astronomical adaptive optics. 1: Modal control optimization. *Astronomy and Astrophysics*, 291, pp. 337–347.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodman, J.W. (2005). *Introduction to Fourier optics. 3rd*, vol. 3. Roberts and Company Publishers.

Gray, M. and Le Roux, B. (2012). Ensemble Transform Kalman Filter, a nonstationary control law for complex AO systems on ELTs: theoretical aspects and first simulations results. In: *Adaptive Optics Systems III*, vol. 8447, p. 84471T.

Guyon, O. (2005). Limits of adaptive optics for high-contrast imaging. *The Astrophysical Journal*, 629(1), p. 592.

Guyon, O. (2018). Extreme adaptive optics. *Annual Review of Astronomy and Astrophysics*, 56, pp. 315–355.

Guyon, O. and Males, J. (2017). Adaptive optics predictive control with empirical orthogonal functions (EOFs). *arXiv preprint arXiv:1707.00570*.

Haffert, S.Y., et al. (2021a). Data-driven subspace predictive control: lab and on-sky demonstration. In: *Techniques and Instrumentation for Detection of Exoplanets X*, vol. 11823, p. 118231C. SPIE.

Haffert, S.Y., et al. (2021b). Data-driven subspace predictive control of adaptive optics for high-contrast imaging. *Journal of Astronomical Telescopes, Instruments, and Systems*, 7(2), p. 029001.

Hardy, J.W. (1998). *Adaptive optics for astronomical telescopes*, vol. 16. Oxford University.

Hardy, J., Feinlieb, J., and Wyant, J. (1974). Real-time Phase Correction of Optical Imaging Systems, Digest of Technical Papers. In: *Topical Meeting on Optical Propagation Through Turbulence, sponsored by OSA, Boulder Colo*.

Heritier, C.T. (2019). *Innovative Calibration Strategies for Large Adaptive Telescopes with Pyramid Wave-Front Sensors*. Ph.D. thesis. Aix Marseille Université.

Hickson, P. (2008). Fundamentals of atmospheric and adaptive optics. *The University of British Columbia: Vancouver, BC, Canada*, pp. 1–68.

Janner, M., Fu, J., Zhang, M., and Levine, S. (2019). When to trust your model: Model-based policy optimization. *arXiv preprint arXiv:1906.08253*.

Jovanovic, N., et al. (2015). The Subaru coronagraphic extreme adaptive optics system: enabling high-contrast imaging on solar-system scales. *Publications of the Astronomical Society of the Pacific*, 127(955), p. 890.

Kaipio, J. and Somersalo, E. (2006). *Statistical and computational inverse problems*, vol. 160. Springer Science & Business Media.

Kasper, M., et al. (2020). PCS ? roadmap for exoearth imaging with the ELT. *ESO Messenger*, 182.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kocijan, J., Murray-Smith, R., Rasmussen, C.E., and Girard, A. (2004). Gaussian process model based predictive control. In: *Proceedings of the 2004 American control conference*, vol. 3, pp. 2214–2219. IEEE.

Kolmogorov, A.N. (1991). Dissipation of energy in the locally isotropic turbulence. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434(1890), pp. 15–17.

van Kooten, M., Doelman, N., and Kenworthy, M. (2017). *Performance of AO predictive control in the presence of non-stationary turbulence*. Instituto de Astrofisica de Canarias.

van Kooten, M., Doelman, N., and Kenworthy, M. (2019). Impact of time-variant turbulence behavior on prediction for adaptive optics systems. *JOSA A*, 36(5), pp. 731–740.

van Kooten, M.A., et al. (2022). Predictive wavefront control on Keck II adaptive optics bench: on-sky coronagraphic results. *Journal of Astronomical Telescopes, Instruments, and Systems*, 8(2), p. 029006.

Korkiakoski, V., Vérinaud, C., and Le Louarn, M. (2008). Improving the performance of a pyramid wavefront sensor with modal sensitivity compensation. *Applied optics*, 47(1), pp. 79–87.

Kulcsár, C., et al. (2006). Optimal control, observers and integrators in adaptive optics. *Optics express, 14(17):7464–7476*.

Lagrange, A.M., et al. (2009). A probable giant planet imaged in the β Pictoris disk. VLT/NaCo deep L'-band imaging. *Astronomy and Astrophysics*, 493(2), pp. L21–L25. doi:10.1051/0004-6361:200811325.

Landman, R. and Haffert, S.Y. (2020). Nonlinear wavefront reconstruction with convolutional neural networks for Fourier-based wavefront sensors. *Opt. Express*, 28(11), pp. 16644–16657. doi:10.1364/OE.389465, url: `http://www.opticsexpress.org/abstract.cfm?URI=oe-28-11-16644`.

Landman, R., Haffert, S.Y., Radhakrishnan, V.M., and Keller, C.U. (2020). Self-optimizing adaptive optics control with reinforcement learning. In: *Adaptive Optics Systems VII*, vol. 11448, p. 1144849. SPIE.

Landman, R., Haffert, S.Y., Radhakrishnan, V.M., and Keller, C.U. (2021). Self-optimizing adaptive optics control with reinforcement learning for high-contrast imaging. *Journal of Astronomical Telescopes, Instruments, and Systems*, 7(3), p. 039002.

Liu, X., Morris, T., and Saunter, C. (2019). Using Long Short-Term Memory for Wavefront Prediction in Adaptive Optics. In: *International Conference on Artificial Neural Networks*, pp. 537–542.

Lukin, V.P. (1995). *Atmospheric adaptive optics*. SPIE Press.

Lyot, B. (1939). The study of the solar corona and prominences without eclipses (George Darwin Lecture, 1939). *Monthly Notices of the Royal Astronomical Society*, 99, p. 580.

Maas, A.L., Hannun, A.Y., and Ng, A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In: *Proc. icml*, vol. 30, p. 3.

Macintosh, B., et al. (2015). Discovery and spectroscopy of the young jovian planet 51 Eri b with the Gemini Planet Imager. *Science*, 350(6256), pp. 64–67. doi:10.1126/science.aac5891.

Macintosh, B., et al. (2014). First light of the gemini planet imager. *proceedings of the National Academy of Sciences*, 111(35), pp. 12661–12666.

Madec, P.Y. (2012). Overview of deformable mirror technologies for adaptive optics and astronomy. In: Ellerbroek, B.L., Marchetti, E., and Véran, J.P., eds, *Adaptive Optics Systems III*, vol. 8447, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, p. 844705.

Madec, P.Y. (1999). Control techniques. *Adaptive optics in astronomy*, pp. 131–154.

Males, J.R. and Guyon, O. (2018). Ground-based adaptive optics coronagraphic performance under closed-loop predictive control. *Journal of Astronomical Telescopes, Instruments, and Systems*, 4(1), p. 019001.

Males, J.R., et al. (2018). MagAO-X: project status and first laboratory results. In: *Adaptive Optics Systems VI*, vol. 10703, p. 1070309. SPIE.

Marois, C., et al. (2010). Images of a fourth planet orbiting HR 8799. *Nature*, 468(7327), pp. 1080–1083. doi:10.1038/nature09684.

Mawet, D., et al. (2012). Review of small-angle coronagraphic techniques in the wake of ground-based second-generation adaptive optics systems. In: Clampin, M.C., Fazio, G.G., MacEwen, H.A., and Oschmann, Jacobus M., J., eds, *Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*, vol. 8442, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, p. 844204. SPIE.

Merkle, F., et al. (1989). Successful tests of adaptive optics. *The Messenger*, 58, pp. 1–4.

Mueller, J.L. and Siltanen, S. (2012). *Linear and nonlinear inverse problems with practical applications*. SIAM.

Nagabandi, A., Kahn, G., Fearing, R.S., and Levine, S. (2018). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566.

Neal, R.M. (2012). *Bayesian learning for neural networks*, vol. 118. Springer Science & Business Media.

Noll, R.J. (1976). Zernike polynomials and atmospheric turbulence. *JOsA*, 66(3), pp. 207–211.

Nousiainen, J., Rajani, C., Kasper, M., and Helin, T. (2021). Adaptive optics control using model-based reinforcement learning. *Optics Express*, 29(10), pp. 15327–15344.

Osband, I. (2016). Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout. In: *NIPS workshop on bayesian deep learning*, vol. 192.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Paschall, R.N. and Anderson, D.J. (1993). Linear quadratic Gaussian control of a deformable mirror adaptive optics system with time-delayed measurements. *Applied optics*, 32(31), pp. 6347–6358.

Paszke, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*, pp. 8024–8035.

Por, E.H., et al. (2018). High Contrast Imaging for Python (HCIPy): an open-source adaptive optics and coronagraph simulator. In: *Adaptive Optics Systems VI*, vol. 10703, Proc. SPIE. url: `https://doi.org/10.1117/12.2314407`.

Pou, B., et al. (2022). Adaptive optics control with multi-agent model-free reinforcement learning. *Opt. Express*, 30(2), pp. 2991–3015. doi:10.1364/OE.444099, url: `http://www.osapublishing.org/oe/abstract.cfm?URI=oe-30-2-2991`.

Poyneer, L.A. and Macintosh, B. (2004). Spatially filtered wave-front sensor for high-order adaptive optics. *JOSA A*, 21(5), pp. 810–819.

Poyneer, L.A., Macintosh, B.A., and Véran, J.P. (2007). Fourier transform wavefront control with adaptive prediction of the atmosphere. *JOSA A*, 24(9), pp. 2645–2660.

Ragazzoni, R. and Farinato, J. (1999). Sensitivity of a pyramidic Wave Front sensor in closed loop Adaptive Optics. *Astronomy & Astrophysics*, 350, pp. L23–L26.

Ragazzoni, R. (1996). Pupil plane wavefront sensing with an oscillating prism. *Journal of modern optics*, 43(2), pp. 289–293.

Roddier, F. (1999). Book Review: Adaptive optics in astronomy/Cambridge U Press, 1999. *Irish Astronomical Journal*, 26, p. 171.

Roggemann, M.C., Welsh, B.M., and Hunt, B.R. (1996). *Imaging through turbulence*. CRC press.

Shatokhina, I., Hutterer, V., and Ramlau, R. (2020). Review on methods for wavefront reconstruction from pyramid wavefront sensor data. *Journal of Astronomical Telescopes, Instruments, and Systems*, 6(1), p. 010901.

Sinquin, B., et al. (2020). On-sky results for adaptive optics control with data-driven models on low-order modes. *Monthly Notices of the Royal Astronomical Society*, 498(3), pp. 3228–3240.

Sun, Z., et al. (2017). A Bayesian regularized artificial neural network for adaptive optics forecasting. *Optics Communications*, 382, pp. 519–527.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. Massachusetts Institute of Technology.

Swanson, R., et al. (2018). Wavefront reconstruction and prediction with convolutional neural networks. In: *Adaptive Optics Systems VI*, vol. 10703, pp. 481–490. SPIE.

Swanson, R., et al. (2021). Closed loop predictive control of adaptive optics systems with convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, 503(2), pp. 2944–2954.

Tatarski, V.I. (2016). *Wave propagation in a turbulent medium*. Courier Dover Publications.

Tyson, R.K. and Frazier, B.W. (2022). *Principles of adaptive optics*. CRC press.

Vérinaud, C. (2004). On the nature of the measurements provided by a pyramid wavefront sensor. *Optics Communications*, 233(1-3), pp. 27–38.

Williams, C.K. and Rasmussen, C.E. (2006). *Gaussian processes for machine learning*, vol. 2, 3. MIT press Cambridge, MA.

Wong, A.P., et al. (2021). Predictive control for adaptive optics using neural networks. *Journal of Astronomical Telescopes, Instruments, and Systems*, 7(1), p. 019001.

Publication I

Nousiainen, J., Rajani, C., Kasper, M., & Helin, T.
Adaptive optics control using model-based reinforcement learning

Reprinted with permission from
Optics Express
29(10), 15327-15344, 2021

Optics EXPRESS

Adaptive optics control using model-based reinforcement learning

JALO NOUSIAINEN,[1,2,*] CHANG RAJANI,[3] MARKUS KASPER,[2] AND TAPIO HELIN[1]

[1] *Department of Computational and Process Engineering, Lappeenranta–Lahti University of Technology, Finland*
[2] *European Southern Observatory, Karl-Schwarzschild-Str. 2, 85748 Garching bei München, Germany*
[3] *Department of Computer Science, University of Helsinki, Finland*
[*] *jalo.nousiainen@lut.fi*

Abstract: Reinforcement learning (RL) presents a new approach for controlling adaptive optics (AO) systems for Astronomy. It promises to effectively cope with some aspects often hampering AO performance such as temporal delay or calibration errors. We formulate the AO control loop as a model-based RL problem (MBRL) and apply it in numerical simulations to a simple Shack-Hartmann Sensor (SHS) based AO system with 24 resolution elements across the aperture. The simulations show that MBRL controlled AO predicts the temporal evolution of turbulence and adjusts to mis-registration between deformable mirror and SHS which is a typical calibration issue in AO. The method learns continuously on timescales of some seconds and is therefore capable of automatically adjusting to changing conditions.

1. Introduction

Atmospheric turbulence distorts astronomical imagery obtained with ground-based telescopes. Adaptive optics (AO) [1–3] is a technique that aims at minimizing the distortions caused by the turbulence. In AO, a wavefront emitted by an astronomical object, such as a star, and distorted by the atmosphere is directed to one or more deformable mirrors (DM) before it propagates to the scientific camera. The distortions are measured with a wavefront sensor (WFS), and optimal image quality is obtained by setting the DM to a shape that partially cancels the distortions after reflection. In this work, we consider the classical single conjugated AO (SCAO) system, which requires a bright star that is close to an object of interest. This reference star is used to calculate distortion caused by the atmosphere along the propagation path. Since the atmosphere is continuously evolving, the mirror's shape has to be controlled in real-time, often from 300 to more than 1000 times a second.

Most AO systems run in a closed-loop configuration, where the WFS measures the wavefront distortions after DM correction (see Fig. 1). The goal of such a control-loop is to minimize the distortions in the measured wavefront; i.e., the residual wavefront. For high contrast imaging (HCI) the wavefront error budget (within the AO controlled region) is often dominated by the temporal delay error [4]. Also real systems often suffer from a dynamic mis-alignment between DM and WFS called mis-registration [5]. Reinforcement learning (RL) provides an automated approach for control, which promises to cope with these limitations of current AO systems. Unlike the classical control methods, RL methods aim to learn a successful closed-loop control strategy via interacting with the system. Hence they do not require accurate models of the components in the control loop and adapt to a changing environment.

In recent years, the merger of RL and deep neural networks (NN), called deep RL, has become increasingly popular due to its effectiveness in problems with large state- and action-spaces. This type of RL has been used, for example, to play video- and board-games on a superhuman level [6,7] and for vision-based real-world robot control [8,9]. Much of the success can be specifically

#420270 https://doi.org/10.1364/OE.420270
Journal © 2021 Received 21 Jan 2021; revised 20 Mar 2021; accepted 22 Apr 2021; published 4 May 2021

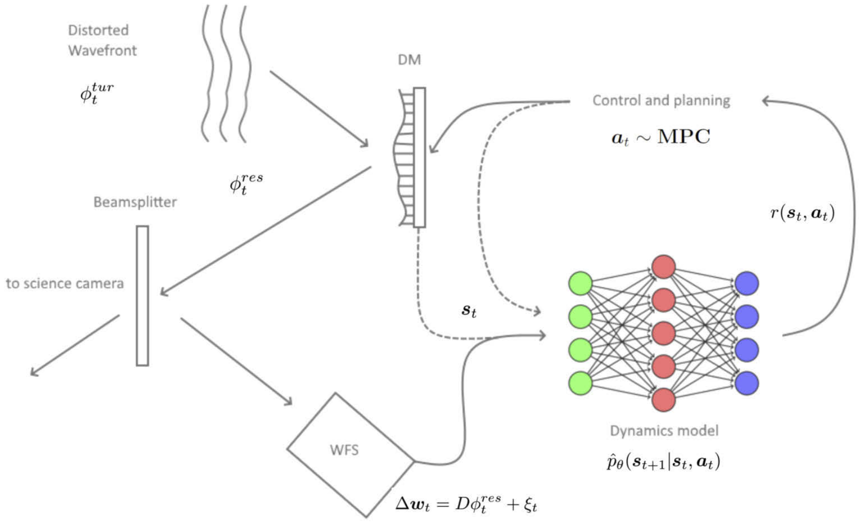


Fig. 1. Overview of the task and method. The distorted wavefront is propagated into the deformable mirror (DM), which is controlled by our control algorithm. The algorithm reads the wavefront sensor (WFS) input, simulates how it will evolve using the learned dynamics model, and plans for the next DM commands with a process called model predictive control (MPC).

attributed to *model-based* RL (MBRL), where a model of the environment is learned using data obtained by interaction, and a *planning algorithm* is used in conjunction to decide the next action. Inspired by these successes, we attempt to generalize the adaptive optics problem to the general framework of reinforcement learning and apply existing algorithms in solving it.

Our starting point is to formulate the closed-loop AO system as a Markov decision process (MDP), the prevailing mathematical framework for reinforcement learning [10]. We describe the state of the AO system as a finite time series of past control voltages and WFS measurements and assume that such a state exhibits Markovian statistics to a good approximation; i.e., each state depends only on the previous state, where a state can also include data from several timesteps from the past. The key to successful prediction lies in finding a reliable model for the system dynamics. Here, we parameterize the dynamics model describing the conditional distribution of the next state given the current state and action using standard NN architectures. This parameterization is fitted to closed loop data in a process called training. Using this framework, we adapt a standard state-of-the-art MBRL algorithm, Probabilistic Ensemble Trajectory Sampling (PETS) [11], to train the model and optimize for the next action; i.e., the set of control voltages.

The paper's structure is as follows: In Section 2, we state the novelty of our method and position our work with respect to existing literature. In Section 3, we give a short description of an AO control loop and the baseline method. Section 4 describes MDP formulation of the AO control loop, setting a platform for RL. Further, we describe the algorithm used and how we adapt it to AO. For small details and a general more in-depth justification of the method, the authors strongly encourage the reader to have a look at the original paper on the algorithm [11]. In Section 5, we demonstrate the performance of our method, through simulation of a small and simple SCAO system controlled either by RL or by the baseline integrator controller. The algorithm and MDP formulation presented is only one way to solve the control loop with RL but already hints at the great potential of MBRL control for AO. Finding the optimal formulation and

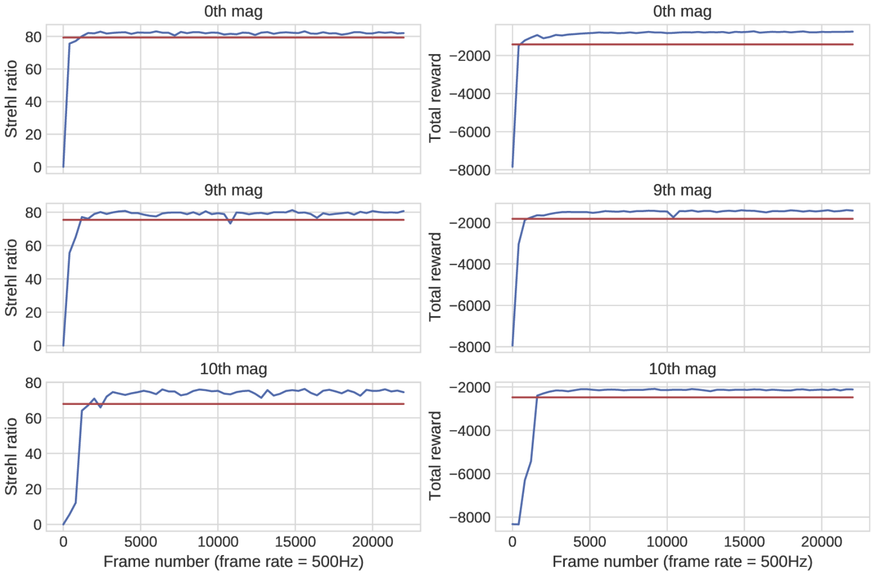


Fig. 2. Learning curves for the proposed RL method. Each learning curve represents the episode performance in H-band Strehl ratios (left) or total reward (right), defined as the sum of rewards at each frame. The red lines are the mean performance of the integrator (Section 3.) and blue lines the performance of PETS after each episode. The learning process itself tries to maximize the total reward and the corresponding improvement in Strehl ratios is a consequence of this process. Our model converges in around 10 episodes, or 4000 frames. The performance level of the integrator control is passed already in after 4 episodes; i.e., 1600 frames.

bringing the computational time of the controller to the required level are left for future research. Finally, Section 6. discusses the topic, especially how MBRL could be implemented in a real system and how to overcome the significant hurdle of inference time and computational jitter.

2. Related work

In order to mitigate the measurement noise and temporal error, predictive controller methods have been proposed for ground-based adaptive optics. These methods include the Kalman filter-based linear quadratic Gaussian control (LQG) [12,13] and its variants [14–18] and predictive filters operating on separate modal coefficients such as Zernike polynomials or Fourier modes [19–21], which provide up to a factor of 1000 gain in raw point spread function contrast in an idealized simulation environment for an extremely large telescope at very small angular separations and using a very bright AO guide star [20]. The contrast performance gets shallower for larger angular separation, smaller telescopes or fainter stars. More recently, data-based predictive methods have emerged in AO literature. Examples include linear predictive filter methods such as empirical orthogonal functions [22], the low-order linear minimum mean square error predictor [23–25], as well as NN-based methods [26–29].

Many of the existing machine learning-based predictive control methods [26–28] have not been studied in a closed-loop configuration, but in principle, they can be integrated into closed-loop systems by utilizing a so-called pseudo-open loop telemetry [22,30]. These procedures consist roughly of two steps: collecting open-loop wavefront estimates from pseudo-open loop telemetry and learning a predictive filter as a supervised learning task. This procedure assumes accurate knowledge of system time lags and DM response, as well as close to linear behavior of the WFS. As a consequence, the predictive filter will inherit the errors in system calibration. These methods, therefore, learn the temporal evolution of the turbulence from the data but rely on modelling of the system components and interactions between them, which leads to the need for external tuning and re-calibration of the predictive controller to ensure robustness. Moreover, some AO systems operate at framerates which are high enough that usually neglected system dynamics (e.g., the finite response time of the DM) become important. Consequently, a control algorithm suffers from the simplifying assumption of a temporal step-wise response.

In contrast to these methods, we present a technique that learns predictive and noise-robust control straight from the system feedback without the set of prior assumptions mentioned earlier and eliminating the need for accurate calibration or modeling assumptions. Our RL formulation uses a generic neural network (NN) architecture to build the dynamics model. NNs have been applied to various aspects of AO before. The topics vary from open-loop systems to the extraction of Zernike coefficients directly from the images and to non-linear wavefront reconstruction (see [31–35]).

Also RL-based concepts have already been applied to AO. Self-adaptive control has been studied in [36], where a deep learning control model is proposed to mitigate alignment errors in the calibration. Model-free RL methods for wavefront sensorless AO have been studied in [37,38], where the method is compared against stochastic parallel gradient descent providing improved correction speed. Finally, model-free RL for ground-based AO was implemented to control tip and tilt only [39]. The model-free RL method they used learns a policy NN that directly outputs the two values for the tip and tilt mirror given the observations. Such methods often require a large number of interactions with the environment, which increase exponentially with the degrees of freedom to be controlled if no additional measures are taken. In contrast, we control each actuator of a high-order DM via model-based RL, formulate ground-based astronomical AO as a general MBRL task, and discuss its potential benefits. We show that state-of-the-art model-based RL learns a self-calibrating noise-robust predictive control law using only a few seconds of past telemetry data.

3. Adaptive optics and the classical integrator

We first present the adaptive optics task, along with useful notation, and then frame it in the reinforcement learning setting. An overview of the AO control loop is given in Fig. 1. The incoming light ϕ_t^{tur} at the timestep t gets corrected by the DM. After this correction the WFS measures the residual wavefront ϕ_t^{res}. Commonly, a linear relationship between the WFS observation and the residual wavefront is assumed; i.e.,

$$\Delta \boldsymbol{w}^t = D\phi_t^{res} + \xi_t, \tag{1}$$

where $\Delta \boldsymbol{w}^t = (\delta w_1^t, \delta w_2^t, \ldots, \delta w_n^t)$ is the WFS data and D is so-called interaction matrix modelling the WFS measurement and ξ_t is the measurement noise typically composed of photon and detector noise. Depending on the type of WFS, a component δw_i of the residual wavefront can represent; e.g., a wavefront modal coefficient, a wavefront slope or the wavefront phase itself. Classical control algorithms are often modelled by a linear mapping of the WFS measurements $\Delta \boldsymbol{w}$ to the residual DM control voltages $\Delta \boldsymbol{v}$; i.e.,

$$\Delta \boldsymbol{v}^t = C\Delta \boldsymbol{w}^t, \tag{2}$$

where C is so-called reconstruction matrix. To obtain the reconstruction matrix, we decompose the DM on a Karhunen–Loeve (K-L) modal basis. Each mode of the K-L basis has a representation in terms of actuator voltages. This relation is fully determined by a linear map B from voltages to modes. The B matrix is computed by a double diagonalization process, which takes into account the geometrical and statistical properties of the telescope [40]. In the following, we utilize a reconstruction matrix defined by the Moore–Penrose pseudo-inverse

$$C = (DB)^{+}. \tag{3}$$

We truncate the number of K-L modes in B to have a stable inversion and a reasonably low noise amplification by C.

Let us now consider a simple non-predictive control algorithm known as the *integrator law*. At a given timestep t, the WFS measures the residual wavefront. The new control voltages $\tilde{\boldsymbol{v}}^t$ are obtained from

$$\tilde{\boldsymbol{v}}^t = \tilde{\boldsymbol{v}}^{t-1} + gC\Delta\boldsymbol{w}^t, \tag{4}$$

where g is the integrator gain. In order to stabilize the loop, the value of g is often fixed below a value of about 0.5 for a two-step delay system [41]. Large values of g increase the correction bandwidth; i.e., the loop reacts faster. On the other hand, a large gain reduces the control loop's stability margin and amplifies noise propagation. The challenge in classical integrator control is in balancing these two effects to minimize the average error of the method [40].

In the following we denote the vector concatenating the past m control voltages

$$\boldsymbol{V}^m(t) = (\tilde{\boldsymbol{v}}^{t-1}, \tilde{\boldsymbol{v}}^{t-2} \ldots \tilde{\boldsymbol{v}}^{t-m})^\top, \tag{5}$$

and the vector concatenating the past k residual voltages, constructed from the WFS slopes, by

$$\Delta\boldsymbol{V}^k(t) = (C\Delta\boldsymbol{w}^{t-1}, C\Delta\boldsymbol{w}^{t-2}, \ldots, C\Delta\boldsymbol{w}^{t-k})^\top. \tag{6}$$

This quantity merely represents WFS measurements in the voltage space projected on the K-L modal basis defined by B. It does not represent voltages applied to the DM.

On the millisecond time scale of AO operations a big part of turbulence is presumably in frozen flow and the turbulence evolution is predictable to some extend [42]. Control methods that use past telemetry data have shown a great potential both in turbulence prediction and noise reduction [22]. In a closed-loop set-up, these methods would, for example, utilize past control and residual voltages in Eqs. (5) and (6), respectively, to construct a pseudo-open loop data stream used for the prediction. This paper aims to obtain a controller with similar properties but without the need for neither an accurate knowledge of time delay, accurate calibration nor a linear response of the WFS to wavefront errors.

4. Adaptive optics as model-based reinforcement learning

4.1. Markov decision process and the dynamics model

We model the closed-loop adaptive optics control problem as an MDP. An MDP consists of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}(s)$ at the given state s, a set of transition probabilities $p(s_{t+1}|s_t, a_t)$ and a reward function $r(s_t, a_t)$.

In AO, the set of actions consists of different combinations of control voltages, and the state consists of the prevailing atmospheric turbulence and the shape of the mirror during the measurement. In practice, we do not have access to the full state of the AO system; i.e., full turbulence, wind speeds and DM shape. We only partially observe the state through a noisy WFS measurement. Consequently, past observations and actions are still valid information for the prediction of the next observation. To account for partial observation and to ensure the Markovian property of state formulation, we define the state as a sequence of previous voltages and residual voltages derived from WFS measurements:

$$\boldsymbol{s}_t = \begin{pmatrix} \boldsymbol{V}^m(t) \\ \Delta \boldsymbol{V}^k(t) \end{pmatrix}, \tag{7}$$

where we typically choose $k = m$. The state includes data from the previous m (or k) time steps and the reconstruction matrix C. We stress that the residual voltages are not applied to the DM. They are merely a quantity closely related to the residual wavefront through Eq. (6), and which the MBRL control approach (see Section 4.3) will try to minimize. The matrix C must only be chosen such that the residual voltages are well observable by the WFS. It does not have to match the actual registration of DM and WFS precisely and could be given by either a previous calibration or derived from a system model. Moreover, previous studies have shown that the neural network-based wavefront reconstructor benefits from involving a linear control matrix with a non-linear WFS [35], and we observe below that MBRL is robust to errors or perturbations in the reconstruction matrix; see Section 5.4.

The action of the MDP is simply a vector of the changes to the control voltages

$$\boldsymbol{a}_t = \Delta \tilde{\boldsymbol{v}}^t. \tag{8}$$

Let us now represent the true transition probability $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$; i.e., the conditional distribution of the next state (including the next WFS residual) given the current state and action as a parameterized distribution family $\hat{p}_\theta(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. The aim of MBRL is to find the optimal approximative model $\hat{p}_\theta$ given a data set from the real environment. We solve this problem by fitting NNs using straightforward supervised learning, detailed in Section 4.3. In our case, the parameters θ represent the weights of the neural networks. The transition probability approximations $\hat{p}_\theta$ represent our probabilistic dynamics and are hence called the dynamics model. It provides an estimation of the next state (of which only the next WFS measurements are new) from the current state and the control voltages. The dynamics model involves information about the interaction of voltages with WFS measurement as well as the system's temporal evolution, including the turbulent wavefront.

In adaptive optics we aim to minimize the residual wavefront φ^{res} over the the whole time interval. The most natural reward for an AO system would be the Strehl ratio, or for a high contrast imaging (HCI) instrument the contrast obtained. Since we are considering a control system with just one WFS, we can only choose a reward function observable on that specific sensor. We choose a reward for a state-action pair as the residual voltages' negative squared norm corresponding to the next measurement:

$$r(\boldsymbol{s}_t, \boldsymbol{a}_t) = -||C\Delta \boldsymbol{w}^{t+1}||^2. \tag{9}$$

This quantity is proportional to the observable part of the negative norm of the true residual wavefront. The WFS measurement is blind to some modes; e.g., the waffle mode for a Shack-Hartmann Sensor (SHS). We ensure that we do not control these modes by projecting each action; i.e., set of control voltages to the control space. That is,

$$\boldsymbol{a}_t = B^+ B \Delta \tilde{\boldsymbol{v}}^t, \tag{10}$$

where B^+B projects the control voltages onto the control space defined by the K-L modes.

4.2. Model-based reinforcement learning

Now that we have defined the MDP components and the dynamics model, we can outline our MBRL approach. First, we initialize an empty data set, and we initialize the dynamics model parameters θ (the weights of the NN) randomly from a zero-mean Gaussian distribution. Then, we collect our first data set by running the AO loop for a particular time interval (an episode) with random actions (DM control voltages) sampled from a zero-mean Gaussian distribution as well.

After the first episode we have the first data set and use it to train the dynamics model. The training is described in more detail in Section 4.3.

We now have a first reasonable guess for the dynamics model and start to use it during the second episode to find the action that maximizes the expected future reward (minimizes the residual voltages) for a given state. This optimization task is called *planning* and replaces the regulator/controller in classical AO. We detail the methods used for this in Section 4.3.2.

After the second and subsequent episodes, the previous data set is concatenated with the new data and the dynamics model is trained again and updated. When the data set gets sufficiently long, old data is removed to ensure that the NNs are trained on sufficiently fresh data only. The dynamics model is entirely learned from data obtained while running the loop; i.e., during the experiment; no simulation or modeling steps are involved.

4.3. PETS algorithm

We implement the MBRL control for AO approach described above following the PETS algorithm [11]. We use OOMAO [43] to simulate the AO system plant (turbulence, telescope, DM, WFS), and the probabilistic ensemble trajectory sampling (PETS) algorithm replaces the classical reconstruction, control and calibration. The algorithm combines a probabilistic ensemble (PE) neural network dynamics model and model predictive control (MPC) [44] that is based on trajectory sampling (TS). We combine the TS with the cross-entropy method (CEM) as described in Section 4.3.2.

4.3.1. Dynamics model

Our choice of the dynamics model, an ensemble of probabilistic NNs, can model two types of uncertainty. Firstly, it models the uncertainty associated with the predictions; e.g., the stochastic behavior of the turbulence and measurement noise, by outputting a variance estimate in addition to a mean prediction. Secondly, it models the uncertainty associated with the model's parameters by learning an ensemble of bootstrap models. Each model has its unique data set to be trained upon that is bootstrap sampled (a statistics term meaning sampling with replacement) from the whole data set recorded so far [45,46].

In preparation for the experiment, we verified that using an ensemble of NNs leads to a superior correction performance as a single NN. Then, we also ran tests and confirmed that estimating the next state's variance improves the performance compared to a fixed variance. Both measures combined stabilize training by a fair amount and eventually reach a higher reward; i.e., a better correction performance.

Each neural network in the ensemble defines a parameterized distribution family $\hat{p}_\theta(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ satisfying

$$\hat{p}_\theta(s_{t+1}|s_t, a_t) \sim \mathcal{N}(\mu_\theta(s_t, a_t), \sigma_\theta^2(s_t, a_t)), \tag{11}$$

where the mean $\mu_\theta(s_t, a_t)$ and the variance $\sigma_\theta^2(s_t, a_t)$ of the Gaussian field are outputs of a neural network. We train the dynamics model ensemble by maximizing the log-likelihood of a Gaussian for which the parameters are outputs of the neural network model. More specifically, given a dataset of N transitions $\mathcal{D} = \{(s_t^i, a_t^i), s_{t+1}^i\}_{i=1}^N$ we maximize the following objective function

$$\hat{\theta} = \text{arg } \max_\theta \log \prod_{i=1}^{N} \hat{p}_\theta(s_{t+1}^i|s_t^i, a_t^i) \tag{12}$$

where $\hat{p}_\theta$ is given by Eq. (11). Each network that is a part of the ensemble is trained similarly, but with different bootstrap sampled data set from $\mathcal{D}$. Each network is modelled as a convolutional neural network with 2 hidden layers of 8 feature maps each. Both layers are activated by a leaky rectified linear unit (LReLU) [47]. We use the concatenated vector $[s_t, a_t]$ as an input and output the mean and log-scale variance of a normal distribution: the distribution of the next state. The maximization in Eq. (12) is done using an extension of stochastic gradient descent called the Adam algorithm [48]. The neural network hyperparameters (e.g., number of layers, convolutional features maps, activation function used) provided relatively fast implementation and performed well in our experiments. We did not tune them further because of the large number of hyperparameters and that the method was not very sensitive to them. However, moving to more complex numeric simulations or lab experiments, hyperparameters have to be more extensively studied. A full pseudocode is given in Algorithm 1, where $\varnothing$ stands for empty set and $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}^{(\text{new})}$ for concatenation of previous dataset and new data set that was collected during the last episode.

Algorithm 1 PETS for adaptive optics

```
function PETS
    Initialize dataset $\mathcal{D} \leftarrow \varnothing$
    for episode in $1 \ldots$ do
        Initialize dynamics model $\hat{p}_\theta$ randomly
        Train $\hat{p}_\theta$ on $\mathcal{D}$ for $L$ epochs using Eq. (12)
        Record transitions $\mathcal{D}^{(\text{new})} = (s_t, s_{t+1}, a_t, r_t), t \in 1 \ldots T$ by running CEM$(s_0, \hat{p}_\theta)$ in simulator for $T$ timesteps
        Set $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}^{(\text{new})}$
```

4.3.2. Planning control

We use the learned dynamics model to plan for the action; i.e., the mirror commands to apply at each timestep. The goal of the planning algorithm is to optimize a sequence of actions $\{a_t, a_{t+1} \cdots a_{t+T}\}$ such that it maximizes the expected reward inside some planning horizon T [44].

For the AO case, the action a_t taken at timestep t takes one timestep to be executed, and one additional timestep for the corresponding observation to be recorded. Therefore, we are essentially doing planning to minimize the observed wavefront sensor measurements up to s_{t+2}; i.e., we implicitly predict the best control action by the DM at the time of the WFS measurement (two frames into the future in this case). This planning horizon of two steps provides stable control to time delays smaller or equal to 2 frames. On a real AO system the time delay is to some extend stochastic and/or non integer. Therefore, the planning horizon should include the longest time delays that may occur in the control loop. Further, in the presence of DM dynamics the effective planning horizon might be a couple of time steps longer, since the control voltage decision are not fully independent.

Starting at the given initial state, the CEM works as follows. We first sample a trajectory of actions a_t, a_{t+1} from a Gaussian distribution parameterized by some starting μ and σ^2. Next we use the learned dynamics model $\hat{p}_\theta$ to produce a sequence of potential next states given the actions and the initial state; i.e., $s_{t+2} \sim \hat{p}_\theta(s_{t+1}, a_{t+1})$, where $s_{t+1} \sim \hat{p}_\theta(s_t, a_t)$. Since the dynamics model is approximated by an ensemble, these states will include samples trained using different bootstrapped training datasets. The algorithm then chooses the so-called *elites*: actions that produce the best rewards, and recalculates the sampling distribution parameters μ, σ^2 to adjust to the elites using a maximum likelihood estimate. Finally, the mean of the sampling distribution is returned as the best trajectory. Note that in the actual task only the first action is executed, after which another transition is observed, and the algorithm is run again using the new observation

Algorithm 2 Cross-entropy method (CEM) for planning in AO

```
procedure CEM($s'$, $\hat{p}_\theta$)
    $\mu \leftarrow a_{t-1}, \sigma^2 \leftarrow \sigma_0^2$
    for i in $1 \ldots n_{\text{iters}}$ do
        Sample actions $a_{t,\ldots,t+2} \sim \mathcal{N}(\mu, \sigma_0^2)$
        Set $s_t \leftarrow s'$
        for t in $1 \ldots T = 2$ do
            Sample possible next states $s_{t+1} \sim \hat{p}_\theta(s_t, a_t)$
            Observe rewards $r(s_t, a_t)$
        Select elites $\hat{a}_1, \ldots, \hat{a}_{n_{\text{elites}}}$ corresponding the largest rewards
        Update $\mu \leftarrow$ mean($\hat{a}$) and $\sigma^2 \leftarrow$ Var($\hat{a}$)
    return $\mu$
```

as the starting state. This procedure of re-planning at each timestep is often referred as model predictive control (MPC). The full pseudo-code is given in Algorithm 2.

5. Results

5.1. Simulation set-up

In the following numerical simulations, the OOMAO simulator serves as the plant of the control system - it only provides the WFS measurements and receives a vector of the control voltages. The PETS algorithm runs in Python and interacts with the plant via Python/MATLAB interface.

We compare the results against the ones obtained by a well-tuned integrator controller as well as a theoretical controller that suffers neither from time delay nor measurement noise. This theoretical controller is computed from the non-delayed noiseless measurement; i.e., it still contains errors due to the aliasing and uncontrolled high order modes. The same limitation also applies to the MBRL and integrator controllers. The optimum integrator gain is always tuned globally to give the best performance (Strehl ratio) at each simulation set-up [GS magnitude and misregistration (MR)] separately. This is done manually, and typical values were between 0.3-0.6 for our simulation setups.

We simulated an 8m telescope observing a single natural guide star (NGS), equipped with a 23 × 23 SHS, and a 24 × 24 DM with a Fried geometry (actuators on the subaperture corners). The DM actuator influence functions are assumed to be Gaussian with a 45% coupling. Atmospheric turbulence is simulated as a sum of three frozen flow layers with Von Karman power spectra combining a Fried parameter r_0 of 15 cm at 550 nm wavelength. The parameters of the atmosphere are listed in Table 1. The loop is running at a framerate of 500 Hz with a time delay of 2 steps. We pick the simulation parameters to demonstrate three key properties of the proposed method:

- The predictive capacity of the method is shown on a system with a negligible measurement noise (see Figs. 5(a) and 3, and Table 2).
- The robustness of the method against observation noise is shown by observing natural guide stars of different magnitudes (see Fig. 4, 5 and Table 2).
- The self-calibrating property is demonstrated by running the same simulations but introducing (MR) between the WFS and DM (see Fig. 7 and Table 1).

We model MR in calibration by changing the alignment between the WFS and the DM in two different directions and shift amplitudes (see Table 1). All images and contrast plots are calculated at $\lambda = 1.65\mu$m (H-band), and the WFS measures at $\lambda = 551$nm (V-band). Wind

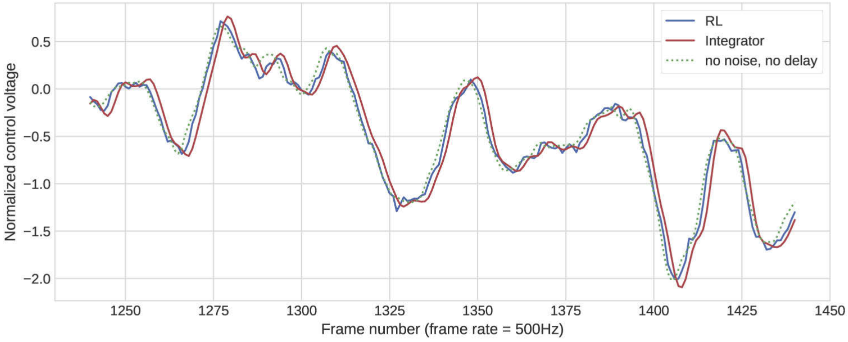


Fig. 3. Predictive control at low-noise (0th mag ngs) regime. We plot a short section of the control voltage time series during the evaluation of each control method: the RL method (blue), the integrator (red), and the theoretical limit of having no noise nor delay (dashed green line). Here we see that the integrator suffers from the time delay, whereas the RL method closely follows the non-delayed signal.

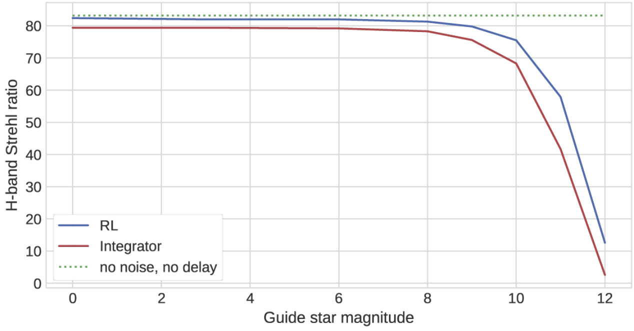
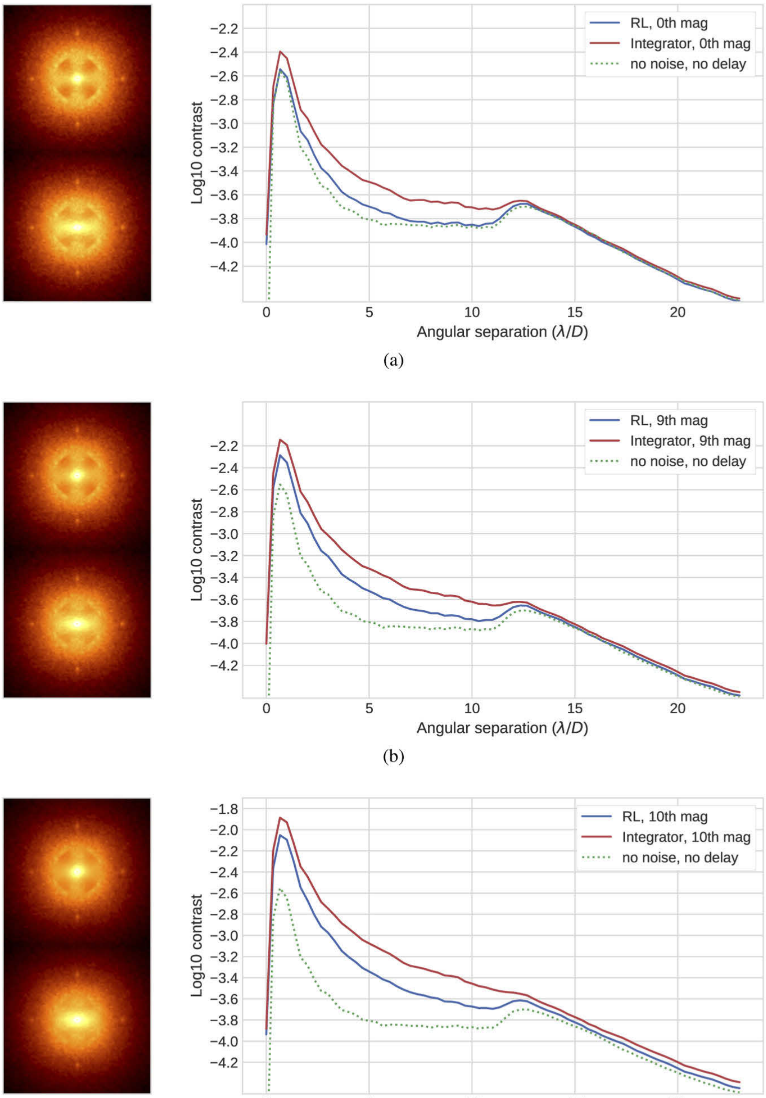


Fig. 4. Results summary. A comparison of H-band Strehl ratios (SR), with respect to star magnitude (GS). The red lines correspond to adapted Integrator and the blue lines the RL method. The dotted green line is the theoretical limit of having no noise nor delay (dashed green line). The RL algorithm always outperforms the integrator and is close to the theoretical limit in the low noise regime.

Table 1. Parameters of the atmosphere

Atmospheric turbulence layers					
	C_N^2 (%)	speed (m/s)	direction (°)	L_0 (m)	altitude (km)
Layer 1	70	15	0	30	0
Layer 2	25	3	45	30	4
Layer 3	5	7.5	90	30	10
Misregistration parameters					
		shift (%)	direction (°)		
Case 1	-	14	225	-	-
Case 2	-	28	135	-	-



(a)

(b)

(c)

Fig. 5. Contrast benefit on three different noise levels. Left: Raw PSF contrast on the pupil plane for RL method (upper panel) and Integrator (lower panel). Right: Azimuthal average of the images. The blue lines are for RL method and red for the integrator. The green dashed line is the contrast obtained with theoretical instantaneous control. The RL method provides a gain in contrast in particular in the direction of the dominant wind. Moreover, in low noise regime, RL provides raw contrast that is close to the theoretical limit of aliasing error.

Table 2. Simulation and MPC parameters

Parameter types	Value	Units
Simulation parameter		
Telescope diameter	8	m
Obstruction ratio	14	percent
Sampling frequency	500	Hz
Active actuators	448	actuators
WFS subapertures	23 × 23	apertures
WFS pixels	10 × 10	pixels
WFS diffraction limited FWHM	2 × 2	pixels
Read-out noise	5	photo-events rms
Photon flux 0/9/10 mag	2033k /511/ 210	photons / frame / lenslet
MPC		
Planning horizon (T)	2	steps
Past DM commands (m)	4	commands
Past WFS measurements (k)	4	frames
CEM elites/particles	200/2000	
CEM iterations	20	
PETS ensemble size	3	

speed and MR are somewhat pessimistic to prevent the error budget from being dominated by the significant aliasing error of the SHS [49].

We set the state of the MDP s_t to include the last four actions and four WFS measurements and set the episode length to 400 frames giving a balance between a fast iteration and a reliable performance estimate. We validate our proposed algorithm by running multiple simulations in the simulator. Each simulation starts with the knowledge of the reconstruction matrix, but zero knowledge of temporal behavior including the time lag. Note here that the sole purpose of the reconstruction matrix is to implement the control space filtering by mapping WFS measurements on residual voltages to be included in the state. We never change it when running the MBRL control, in particular we do not update it to match the MR. Our model learns to compensate for the measurement noise, misregistration in the reconstruction matrix, and the atmosphere's temporal behavior by interacting with the environment.

5.2. Training

To demonstrate how fast the method learns a successful control strategy in different noise conditions and MRs, we compare the learning curve of the method to the baseline of the integrator (see Figs. 2 and 6). In terms of loss; i.e., the negative reward over the episode, our model outperforms the integrator baseline after about 1600 frames and reaches its full potential in about 4000 frames, in all of the test cases. The total loss in the figure corresponds to the sum of normalized residual voltages computed from the WFS measurements. For the simulated system running at 500 Hz, 1600 timesteps is equivalent to 3.2 seconds of actual time, while 4000 is 8 seconds. As described in Section 4.2, we train and update the dynamics model after each episode. The loop is suspended during this time, which amounts for a several seconds given our rather shallow NN architecture and moderate computational power. At the telescope with typically variable observing conditions (wind speed and directions, seeing, guide star magnitudes), the dynamics model has to be trained in parallel to the observation, for example using a separate computer. The available time for training is then set by the episode length and should not exceed the time-scale of environment variability.

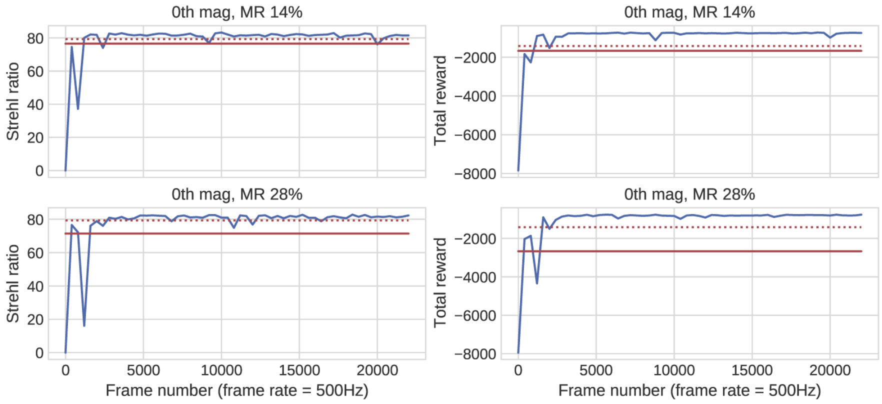


Fig. 6. Learning curves under MR for proposed RL method. The performance vs time of the MBRL controller is shown in blue, and the mean Strehl ratio of the integrator is shown in red (solid: subject to MR, dashed: no MR). Due to the non optimal geometry some high order modes are not visible in the WFS anymore and learning curves contain more variance. Nevertheless, in both cases, the RL algorithm reaches a better performance than the Intergrator with no MR.

5.3. Prediction and noise robustness

We compare the correction performance of the fully converged PETS models to the integrator in terms of raw point spread function (PSF) contrast [50] and Strehl ratio [3]. Each simulation run consists of 8000 frames; i.e., 16 s. The resulting H-band Strehl ratios are computed from the wavefront error maps by Marechal's approximation [3], and are presented in Fig. 4. The MBRL control outperforms the integrator in all cases. A predictive capacity of the MBRL algorithm should result in an improved raw PSF contrast by reducing the notorious wind-driven halo (WDH) [51]. The raw PSF contrast is given by the intensity ratio of the perfect coronagraphic PSF [52] at a certain angular separation over to the peak intensity of the non-coronagraphic image. In Fig. 5(a), we see that the RL method significantly reduces the WDH in all noise cases and hence delivers a better raw PSF contrast especially along the dominant wind direction. We also analyze a time series of one randomly picked actuator shown in Fig. 3, and see that the RL method follows the non-delayed signal much closer than the integrator which exhibits the expected 2-frame delay between incident wavefront and correction by the DM.

5.4. Performance under misregistration

Besides the predictive power, MBRL may provide other benefits for AO. One such benefit could be the automatic adaptation to dynamic MR between DM and WFS. MR is often introduced through mechanically or thermally induced flexure in a real AO system and negatively affects the performance if left uncompensated. Algorithms to detect and compensate for MR exist [5], but combining these with a data-driven predictive control, might not be trivial or at least might need online tuning of hyper-parameters involved. In turn, RL does not make a specific assumption on the origin of error terms. Consequently, altogether the same algorithm with the same hyperparameters, including the reconstruction matrix C, also learns errors due to MR. Prospects are that RL might also learn to minimize some error terms we are not expecting.

In order to verify this claim, we ran a simulation of the bright guide star case while shifting the WFS with respect to the DM by 14% to the upper left (1 px up and 1 px right on the WFS) and in another case by 28% to the lower right (2 px down and 2 px right). Note that the reconstruction matrix C does not include the MR; i.e., the residual voltage presentation of WFS measurement does not match the mirror's voltage presentation anymore. The results are shown in Figs. 6 and 7. The MBRL control maintains its performance and predictive capacity even when a serve MR of 28% of a subaperture is applied. Only at high spatial frequencies close to the DM correction radius [50], we see a small contrast degradation in the 28% MR case. This is due to the non optimal alignment geometry; i.e., some higher order modes on the DM are not anymore visible in the WFS. The RL method also learns to stabilize these modes.

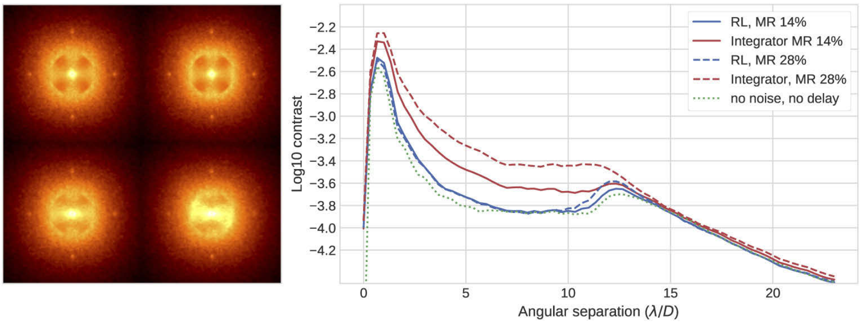


Fig. 7. Predictive control under mis-registration. Left: Raw PSF contrast on the pupil plane. Right: Radial averages over the image. The blue lines are for PETS algorithm and red for the integrator with non corrected MR. The PETS maintains the performance under a sever MR.

6. Discussion

We have formulated the control task of a closed-loop adaptive optics system as a Markov decision process and evaluated the performance of standard deep reinforcement learning algorithms on such a system. Our simulation results demonstrate that a state-of-the-art MBRL algorithm PETS robustly performs well with no environment-specific assumptions, apart from a generic reconstruction matrix. Moreover, the MBRL method predicted the turbulence evolution to a good approximation and automatically adapted to misregistration between DM and WFS, and was robust to measurement noise. Even though the algorithm itself is rather complicated to implement, its usage is simple: the algorithm calibrates, tunes, and maintains itself automatically.

The MBRL method operates on control voltages and residual voltages which are derived from the residual WFS measurements and takes into account closed-loop dynamics along with the temporal evolution of the atmosphere. All the data needed is recorded on the control system itself eliminating dependencies on any numerical simulator or assumptions on the physics of the system. The MBRL control also outperformed classical integrator control in all simulation environments considered in Section 4.3.2. The simulated performance is limited by the aliasing error of the SHS. With our single sensor setup and the objective to null future measurements, the correction of the DM unavoidably includes low spatial frequency aberrations which cancel the SHS signal of high spatial frequency turbulence [53]. Finally, the MBRL method learns quickly requiring only 1600 timesteps in the simulator to surpass the baseline controller and converges at around 4000 timesteps.

We simulate a relatively low order system with 24 actuators across the pupil. On the one hand, this keeps the execution times low with our moderate computational resources. On the other hand, the chosen system size is very relevant, because it simulates the size foreseen for the second AO stages currently planned or under development [54–56] and to be added to already existing first AO stages. While here we consider a single stage SCAO system, our method could be extended to control such 2nd-stage AO by including the first stage's voltages in the state as well.

In future work, we plan to extend the algorithm and comprehensively study a system with more complex DM dynamics, non-linear WFS such as the Pyramid WFS, saturations, alignment errors, turbulence boiling, and a cascaded AO system with a fast second stage. In particular, the future extreme AO systems on the upcoming generation of extremely large telescopes will control more than 10^4 degrees of freedom; as such, scalability of the method shall be considered.

Future work should also address the challenges imposed by a variable turbulence. Understanding the trade-off between model complexity and fast training is essential for a successful implementation. Our MBRL method already learns continuously on a timescales of several seconds. Therefore, prospects are good that it is capable of automatically adjusting to changing conditions on timescales where atmosphere parameters typically change [42]

Finally, we believe that, the biggest and most important challenge for a successful on-sky implementation of MBRL control for AO is the computational complexity of the method. In this work, the computational time at each timestep of the MPC on 448 degrees of freedom is around 80-120 ms using a laptop equipped with a single NVIDIA Quadro RTX 3000 GPU and a straightforward implementation in PyTorch [57]. Both, the delay and the temporal jitter are too large for a stable control of a real system. In contrast to a real system whose cadence is defined by the atmosphere and WFS framerate, our simulations are stepwise and, therefore, not sensitive to the jitter, and no strategies to minimize it was devised. Jitter could, for example, be mitigated by exiting the planning algorithm after a given time rather than after a fixed number of iterations (20 in our simulations).

The large computational cost could be alleviated by reducing the number of parameters in the dynamics model, employing fewer samples in the planning phase, and tuning the CEM procedure's hyperparameters. It seems feasible that these points combined with better hardware and optimized low-level implementation are sufficient to bring the running time of our method with 448 degrees of freedom down into the range needed for an on-sky system.

However, the algorithm's brute force approach could possibly be improved. A promising approach to speed up the MBRL control system, could be to replace the dynamics model and/or the planning algorithm to reduce computational complexity. We proposed a dynamics model composed of an ensemble of convolutional NNs. If the non-linear property of NNs turns out not to be needed, a much simpler linear model; e.g., an autoregressive model, could be used instead. Also, we are already investigating other methods that replace the planning phase of MBRL with a so-called policy function [58], which could be implemented as a NN and therefore avoid iterations and make the controller fast.

Finally, an efficient possible direction to reduce computational effort is to apply MBRL control only to a low-dimensional subset of the controlled parameters. For example, modal control could allow us to control a small set of modes with MBRL, while other modes are controlled classically.

Funding. European Southern Observatory; Luonnontieteiden ja Tekniikan Tutkimuksen Toimikunta (326961).

Acknowledgments. We thank Arto Klami, Cedric Heritier, and Miska Le Louarn for fruitful discussions and guidance on the project.

Disclosures. The authors declare no conflicts of interest.

References

1. H. W. Babcock, "The possibility of compensating astronomical seeing," Publ. Astron. Soc. Pac. **65**, 229–236 (1953).
2. J. W. Hardy, *Adaptive optics for astronomical telescopes*, vol. 16 (Oxford University, 1998).
3. F. Roddier, *Adaptive optics in astronomy* (Cambridge University, 1999).
4. O. Guyon, "Limits of adaptive optics for high-contrast imaging," ApJ **629**(1), 592–614 (2005).

5. C. Heritier, S. Esposito, T. Fusco, B. Neichel, S. Oberti, R. Briguglio, G. Agapito, A. Puglisi, E. Pinna, and P. Y. Madec, "A new calibration strategy for adaptive telescopes with pyramid wfs," Mon. Not. R. Astron. Soc. **481**(2), 2829–2840 (2018).
6. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602 (2013).
7. D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," arXiv preprint arXiv:1712.01815 (2017).
8. F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," arXiv preprint arXiv:1511.03791 (2015).
9. D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," arXiv preprint arXiv:1806.10293 (2018).
10. R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (Massachusetts Institute of Technology, 2018).
11. K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, (2018), pp. 4754–4765.
12. C. Kulcsár, H.-F. Raynaud, C. Petit, J.-M. Conan, and P. V. D. Lesegno, "Optimal control, observers and integrators in adaptive optics," Opt. Express **14**(17), 7464–7476 (2006).
13. R. N. Paschall and D. J. Anderson, "Linear quadratic gaussian control of a deformable mirror adaptive optics system with time-delayed measurements," Appl. Opt. **32**(31), 6347–6358 (1993).
14. M. Gray and B. Le Roux, "Ensemble transform kalman filter, a nonstationary control law for complex ao systems on elts: theoretical aspects and first simulations results," in *Adaptive Optics Systems III*, vol. 8447 (International Society for Optics and Photonics, 2012), p. 84471T.
15. J.-M. Conan, H. Raynaud, C. A. R. Kulcsár, S. Meimon, and G. Sivo, "Are integral controllers adapted to the new era of elt adaptive optics?" in *AO4ELT*, (2011).
16. C. Correia, J.-M. Conan, C. Kulcsár, H.-F. Raynaud, and C. Petit, "Adapting optimal lqg methods to elt-sized ao systems," in *1st AO4ELT conference-Adaptive Optics for Extremely Large Telescopes*, (EDP Sciences, 2010), p. 07003.
17. C. Correia, H.-F. Raynaud, C. Kulcsár, and J.-M. Conan, "On the optimal reconstruction and control of adaptive optical systems with mirror dynamics," J. Opt. Soc. Am. A **27**(2), 333–349 (2010).
18. C. M. Correia, C. Z. Bond, J.-F. Sauvage, T. Fusco, R. Conan, and P. L. Wizinowich, "Modeling astronomical adaptive optics performance with temporally filtered wiener reconstruction of slope data," J. Opt. Soc. Am. A **34**(10), 1877–1887 (2017).
19. L. A. Poyneer, B. A. Macintosh, and J.-P. Véran, "Fourier transform wavefront control with adaptive prediction of the atmosphere," J. Opt. Soc. Am. A **24**(9), 2645–2660 (2007).
20. J. R. Males and O. Guyon, "Ground-based adaptive optics coronagraphic performance under closed-loop predictive control," Telescopes, Instruments, and Systems J. Astron. Telesc. Instrum. Syst. **4**(01), 1 (2018).
21. C. Dessenne, P.-Y. Madec, and G. Rousset, "Optimization of a predictive controller for closed-loop adaptive optics," Appl. Opt. **37**(21), 4623–4633 (1998).
22. O. Guyon and J. Males, "Adaptive optics predictive control with empirical orthogonal functions (eofs)," arXiv preprint arXiv:1707.00570 (2017).
23. M. van Kooten, N. Doelman, and M. Kenworthy, *Performance of AO predictive control in the presence of non-stationary turbulence* (Instituto de Astrofisica de Canarias, 2017).
24. M. van Kooten, N. Doelman, and M. Kenworthy, "Impact of time-variant turbulence behavior on prediction for adaptive optics systems," J. Opt. Soc. Am. A **36**(5), 731–740 (2019).
25. S. Y. Haffert, J. R. Males, L. M. Close, K. V. Gorkom, J. D. Long, A. D. Hedglen, O. Guyon, L. Schatz, M. Kautz, J. Lumbres, A. Rodack, J. M. Knight, H. Sun, and K. Fogarty, "Data-driven subspace predictive control of adaptive optics for high-contrast imaging," (2021).
26. R. Swanson, M. Lamb, C. Correia, S. Sivanandam, and K. Kutulakos, "Wavefront reconstruction and prediction with convolutional neural networks," in *Adaptive Optics Systems VI*, vol. 10703 (International Society for Optics and Photonics, 2018), p. 107031F.
27. X. Liu, T. Morris, and C. Saunter, "Using long short-term memory for wavefront prediction in adaptive optics," in *International Conference on Artificial Neural Networks*, (Springer, 2019), pp. 537–542.
28. Z. Sun, Y. Chen, X. Li, X. Qin, and H. Wang, "A bayesian regularized artificial neural network for adaptive optics forecasting," Opt. Commun. **382**, 519–527 (2017).
29. P. C. McGuire, D. G. Sandler, M. Lloyd-Hart, and T. A. Rhoadarmer, "Adaptive optics: Neural network wavefront sensing, reconstruction, and prediction," in *Scientific Applications of Neural Nets*, (Springer, 1999), pp. 97–138.
30. R. Jensen-Clem, C. Z. Bond, S. Cetre, E. McEwen, P. Wizinowich, S. Ragland, D. Mawet, and J. Graham, "Demonstrating predictive wavefront control with the keck ii near-infrared pyramid wavefront sensor," in *Techniques and Instrumentation for Detection of Exoplanets IX*, vol. 11117 (International Society for Optics and Photonics, 2019), p. 111170W.

31. S. L. S. Gómez, C. González-Gutiérrez, E. D. Alonso, J. D. Santos, M. L. S. Rodríguez, T. Morris, J. Osborn, A. Basden, L. Bonavera, J. G.-N. González, and F. J. de Cos Juez, "Experience with artificial neural networks applied in multi-object adaptive optics," Publ. Astron. Soc. Pac. **131**(1004), 108012 (2019).
32. J. Osborn, D. Guzman, F. J. de Cos Juez, A. G. Basden, T. J. Morris, E. Gendron, T. Butterley, R. M. Myers, A. Guesalaga, F. Sanchez Lasheras, M. Gomez Victoria, M. L. Sánchez Rodríguez, D. Gratadour, and G. Rousset, "Open-loop tomography with artificial neural networks on canary: on-sky results," Mon. Not. R. Astron. Soc. **441**(3), 2508–2514 (2014).
33. C. González-Gutiérrez, J. D. Santos, M. Martínez-Zarzuela, A. G. Basden, J. Osborn, F. J. Díaz-Pernas, and F. J. de Cos Juez, "Comparative study of neural network frameworks for the next generation of adaptive optics systems," Sensors **17**(6), 1263 (2017).
34. D. Sandler, T. Barrett, D. Palmer, R. Fugate, and W. Wild, "Use of a neural network to control an adaptive optics system for an astronomical telescope," Nature **351**(6324), 300–302 (1991).
35. R. Landman and S. Y. Haffert, "Nonlinear wavefront reconstruction with convolutional neural networks for fourier-based wavefront sensors," Opt. Express **28**(11), 16644–16657 (2020).
36. Z. Xu, P. Yang, K. Hu, B. Xu, and H. Li, "Deep learning control model for adaptive optics systems," Appl. Opt. **58**(8), 1998–2009 (2019).
37. H. Ke, B. Xu, Z. Xu, L. Wen, P. Yang, S. Wang, and L. Dong, "Self-learning control for wavefront sensorless adaptive optics system through deep reinforcement learning," Optik **178**, 785–793 (2019).
38. K. Hu, Z. Xu, W. Yang, and B. Xu, "Build the structure of wfsless ao system through deep reinforcement learning," IEEE Photonics Technol. Lett. **30**(23), 2033–2036 (2018).
39. R. Landman, S. Y. Haffert, V. M. Radhakrishnan, and C. U. Keller, "Self-optimizing adaptive optics control with reinforcement learning," in *Adaptive Optics Systems VII*, vol. 11448 (International Society for Optics and Photonics, 2020), p. 1144849.
40. E. Gendron and P. Léna, "Astronomical adaptive optics. 1: Modal control optimization," Astron. Astrophys. **291**, 337–347 (1994).
41. P.-Y. Madec, "Control techniques," Adaptive optics in astronomy pp. 131–154 (1999).
42. L. Poyneer, M. van Dam, and J.-P. Véran, "Experimental verification of the frozen flow atmospheric turbulence assumption with use of astronomical adaptive optics telemetry," J. Opt. Soc. Am. A **26**(4), 833–846 (2009).
43. R. Conan and C. Correia, "Object-oriented matlab adaptive optics toolbox," in *Adaptive optics systems IV*, vol. 9148 (International Society for Optics and Photonics, 2014), p. 91486C.
44. E. F. Camacho and C. B. Alba, *Model predictive control* (Springer Science & Business Media, 2013).
45. B. Efron and R. J. Tibshirani, *An introduction to the bootstrap* (CRC press, 1994).
46. B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," arXiv preprint arXiv:1612.01474 (2016).
47. A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30 (2013), p. 3.
48. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).
49. F. J. Rigaut, J.-P. Veran, and O. Lai, "Analytical model for Shack-Hartmann-based adaptive optics systems," in *Adaptive Optical System Technologies*, vol. 3353 D. Bonaccini and R. K. Tyson, eds., International Society for Optics and Photonics (SPIE, 1998), pp. 1038–1048.
50. M. D. Perrin, A. Sivaramakrishnan, R. B. Makidon, B. R. Oppenheimer, and J. R. Graham, "The structure of high strehl ratio point-spread functions," ApJ **596**(1), 702–712 (2003).
51. F. Cantalloube, E. H. Por, K. Dohlen, J.-F. Sauvage, A. Vigan, M. Kasper, N. Bharmal, T. Henning, W. Brandner, J. Milli, C. Correia, and T. Fusco, "Origin of the asymmetry of the wind driven halo observed in high-contrast images," ApJ **620**, L10 (2018).
52. C. Cavarroc, A. Boccaletti, P. Baudoz, T. Fusco, and D. Rouan, "Fundamental limitations on earth-like planet detection with extremely large telescopes," ApJ **447**(1), 397–403 (2006).
53. J.-P. Veran, F. J. Rigaut, H. Maitre, and D. Rouan, "Estimation of the adaptive optics long-exposure point spread function using control loop data: recent developments," in *Adaptive Optics and Applications*, vol. 3126 (International Society for Optics and Photonics, 1997), pp. 81–92.
54. A. Boccaletti, G. Chauvin, D. Mouillet, O. Absil, F. Allard, S. Antoniucci, J.-C. Augereau, P. Barge, A. Baruffolo, J.-L. Baudino, P. Baudoz, M. Beaulieu, M. Benisty, J.-L. Beuzit, A. Bianco, B. Biller, B. Bonavita, M. Bonnefoy, S. Bos, J.-C. Bouret, W. Brandner, N. Buchschache, B. Carry, F. Cantalloube, E. Cascone, A. Carlotti, B. Charnay, A. Chiavassa, E. Choquet, Y. Clenet, A. Crida, J. De Boer, V. De Caprio, S. Desidera, J.-M. Desert, J.-B. Delisle, P. Delorme, K. Dohlen, D. Doelman, C. Dominik, V. Orazi, C. Dougados, S. Doute, D. Fedele, M. Feldt, F. Ferreira, C. Fontanive, T. Fusco, R. Galicher, A. Garufi, E. Gendron, A. Ghedina, C. Ginski, J.-F. Gonzalez, D. Gratadour, R. Gratton, T. Guillot, S. Haffert, J. Hagelberg, T. Henning, E. Huby, M. Janson, I. Kamp, C. Keller, M. Kenworthy, P. Kervella, Q. Kral, J. Kuhn, E. Lagadec, G. Laibe, M. Langlois, A.-M. Lagrange, R. Launhardt, L. Leboulleux, H. Le Coroller, G. Li Causi, M. Loupias, A. Maire, G. Marleau, F. Martinache, P. Martinez, D. Mary, M. Mattioli, J. Mazoyer, H. Meheut, F. Menard, D. Mesa, N. Meunier, Y. Miguel, J. Milli, M. Min, P. Molliere, C. Mordasini, G. Moretto, L. Mugnier, G. Muro Arena, N. Nardetto, M. N. Diaye, N. Nesvadba, F. Pedichini, P. Pinilla, E. Por, A. Potier, S. Quanz, J. Rameau, R. Roelfsema, D. Rouan, E. Rigliaco, B. Salasnich, M. Samland, J.-F. Sauvage, H.-M. Schmid, D. Segransan, I. Snellen, F. Snik, F. Soulez, E. Stadler, D. Stam, M. Tallon, P. Thebault, E. Thiebaut, C.

Tschudi, S. Udry, R. van Holstein, P. Vernazza, F. Vidal, A. Vigan, R. Waters, F. Wildi, M. Willson, A. Zanutta, A. Zavagno, and A. Zurlo, "Sphere+: Imaging young jupiters down to the snowline," arXiv preprint arXiv:2003.05714 (2020).

55. B. Chazelas, C. Lovis, N. Blind, J. Kühn, L. Genolet, I. Hughes, M. Turbet, J. Hagelberg, N. Restori, M. Kasper, and N. N. C. Urra, "Ristretto: a pathfinder instrument for exoplanet atmosphere characterization," in *Adaptive Optics Systems VII*, vol. 11448 (International Society for Optics and Photonics, 2020), p. 1144875.
56. M. Kasper, N. C. Urra, P. Pathak, M. Bonse, J. Nousiainen, B. Engler, C. T. Heritier, J. Kammerer, S. Leveratto, C. Rajani, P. Bristow, M. Le Louarn, P.-Y. Madec, S. Ströbele, C. Verinaud, A. Glauser, S. P. Quanz, T. Helin, C. Keller, F. Snik, A. Boccaletti, G. Chauvin, D. Mouillet, C. Kulcsár, and H.-F. Raynaud, "Pcs – roadmap for exoearth imaging with the elt," ESO Messenger **182**, 38–43 (2021).
57. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, (2019), pp. 8024–8035.
58. M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, (Citeseer, 2011), pp. 465–472.

Publication II

Nousiainen, J., Rajani, C., Kasper, M., Helin, T., Haffert, S. Y., Vèrinaud, C., Males, J. R. , Van Gorkom, K. V., Close, L. M. , Long, J. D., Hedglen, A. D., Guyon, O., Schatz, L., Kautz, M., Lumbres, J., Rodack, A., Knight, J. M. & Miller, K.

Toward on-sky adaptive optics control using reinforcement learning

Reprinted with permission from
Astronomy & Astrophysics
664, A71, 2022.

A&A 664, A71 (2022)
https://doi.org/10.1051/0004-6361/202243311
© J. Nousiainen et al. 2022

Astronomy & Astrophysics

Toward on-sky adaptive optics control using reinforcement learning

Model-based policy optimization for adaptive optics

J. Nousiainen[1,3], C. Rajani[2], M. Kasper[3], T. Helin[1], S. Y. Haffert[4,⋆], C. Vérinaud[3], J. R. Males[4], K. Van Gorkom[4], L. M. Close[4], J. D. Long[4], A. D. Hedglen[4,5], O. Guyon[4,5,6,7], L. Schatz[8], M. Kautz[4,5], J. Lumbres[4,5], A. Rodack[4,5], J. M. Knight[4,5], and K. Miller[8]

[1] Lappeenranta–Lahti University of Technology, Lappeenranta, Finland
e-mail: jalo.nousiainen@lut.fi
[2] University of Helsinki, Department of Computer Science, Helsinki, Finland
[3] European Southern Observatory, Garching bei München, Germany
[4] University of Arizona, Steward Observatory, Tucson, Arizona, USA
[5] Wyant College of Optical Science, University of Arizona, 1630 E University Blvd, Tucson, AZ 85719, USA
[6] Astrobiology Center, National Institutes of Natural Sciences, 2-21-1 Osawa, Mitaka, Tokyo, JAPAN
[7] National Astronomical Observatory of Japan, Subaru Telescope, National Institutes of Natural Sciences, Hilo, HI 96720, USA
[8] Kirtland Air Force Base, Air Force Research Laboratory, Albuquerque, NM, USA

Received 11 February 2022 / Accepted 25 May 2022

ABSTRACT

Context. The direct imaging of potentially habitable exoplanets is one prime science case for the next generation of high contrast imaging instruments on ground-based, extremely large telescopes. To reach this demanding science goal, the instruments are equipped with eXtreme Adaptive Optics (XAO) systems which will control thousands of actuators at a framerate of kilohertz to several kilohertz. Most of the habitable exoplanets are located at small angular separations from their host stars, where the current control laws of XAO systems leave strong residuals.
Aims. Current AO control strategies such as static matrix-based wavefront reconstruction and integrator control suffer from a temporal delay error and are sensitive to mis-registration, that is, to dynamic variations of the control system geometry. We aim to produce control methods that cope with these limitations, provide a significantly improved AO correction, and, therefore, reduce the residual flux in the coronagraphic point spread function (PSF).
Methods. We extend previous work in reinforcement learning for AO. The improved method, called the Policy Optimization for Adaptive Optics (PO4AO), learns a dynamics model and optimizes a control neural network, called a policy. We introduce the method and study it through numerical simulations of XAO with Pyramid wavefront sensor (PWFS) for the 8-m and 40-m telescope aperture cases. We further implemented PO4AO and carried out experiments in a laboratory environment using Magellan Adaptive Optics eXtreme system (MagAO-X) at the Steward laboratory.
Results. PO4AO provides the desired performance by improving the coronagraphic contrast in numerical simulations by factors of 3–5 within the control region of deformable mirror and PWFS, both in simulation and in the laboratory. The presented method is also quick to train, that is, on timescales of typically 5–10 s, and the inference time is sufficiently small (<ms) to be used in real-time control for XAO with currently available hardware even for extremely large telescopes.

Key words. instrumentation: high angular resolution – instrumentation: adaptive optics – atmospheric effects – methods: data analysis – techniques: high angular resolution – methods: numerical

1. Introduction

The study of extrasolar planets (exoplanets) and exoplanetary systems is one of the most rapidly developing fields of modern astrophysics. More than 3000 confirmed exoplanets have been identified mainly through indirect methods by NASA's *Kepler* mission[1]. High-contrast imaging (HCI) detections are mostly limited to about a dozen very young and luminous giant exoplanets (e.g., Marois et al. 2010; Lagrange et al. 2009; Macintosh et al. 2015) due to the challenging contrast requirements at a fraction of an arcsecond angular distance from the star which could be a billion times brighter than the exoplanet.

High-contrast imaging aims to separate the exoplanet light from stellar light optically, thereby dramatically increasing the signal-to-noise ratio (S/N) over the one provided by indirect methods. However, significant advances in HCI technology are needed to address two major scientific questions: the architectures of outer planetary systems, which remain essentially unexplored (e.g., Dressing & Charbonneau 2015; Fernandes et al. 2019); and the atmospheric composition of small exoplanets outside the solar system, which is especially interesting because it addresses the question of habitability and life in the universe.

⋆ NASA *Hubble* Fellow
[1] Exoplanet Orbit Database: http://exoplanets.org/

For ground-based observations, HCI combines eXtreme Adaptive Optics (XAO, e.g., Guyon 2005, 2018) and coronagraphy (Mawet et al. 2012) with a way to distinguish stellar quasi-static speckles (QSS) produced by imperfect instrument optics from the exoplanet such as spectral and angular differential imaging (Marois et al. 2004, 2006) or high-dispersion spectroscopy (Snellen et al. 2015). With an optimized instrument design, the XAO residual halo may be the dominant source of noise (Otten et al. 2021). Therefore, minimizing the XAO residuals is a key objective for ground-based HCI.

Adaptive optics systems typically run in a closed-loop configuration, where the wavefront sensor (WFS) measures the wavefront distortions after deformable mirror (DM) correction. The objective of this control loop is to minimize the distortions in the measured wavefront, that is, the residual wavefront, which, in theory, corresponds to minimizing the speckle intensity in the post-coronagraphic image. In the case of a widely used integrator controller, temporal delay error and photon noise usually dominate the wavefront error budget in the spatial frequency regime controlled by the DM (Guyon 2005; Fusco et al. 2006). A big part of the turbulence is presumably in frozen flow considering the millisecond timescale of AO control, and hence a significant fraction of wavefront disturbances can be predicted (Poyneer et al. 2009). Therefore, control methods that use past telemetry data have shown a significant potential for reducing the temporal error and photon noise (Males & Guyon 2018; Guyon & Males 2017; Correia et al. 2020). Further, real systems suffer from dynamic modeling errors such as misregistration (Heritier et al. 2018), optical gain effect for the Pyramid WFS (Korkiakoski et al. 2008; Deo et al. 2019), and temporal jitter (Poyneer & Véran 2008). Combined, these errors lead to a need for external tuning and recalibration of a standard pseudo-open-loop predictive controller to ensure robustness.

An up-and-coming field of research aimed at improving AO control methods is the application of fully data-driven control methods, where the control voltages are separately added to the learned control model (Nousiainen et al. 2021; Landman et al. 2020, 2021; Haffert et al. 2021a,b; Pou et al. 2022). A significant benefit of fully data-driven control in closed-loop is that it does not require an estimate of the system's open-loop temporal evolution and that it is, therefore, insensitive to pseudo-open-loop reconstruction errors, such as the optical gain effect (Haffert et al. 2021a). In particular, reinforcement learning (RL) has also been shown to cope with temporal and misregistration errors (Nousiainen et al. 2021). RL is an active branch of machine learning that learns a control task via interaction with the environment. The principal idea is to let the method feed actions to the environment, observe the outcome, and then improve the control strategy regarding the long-term reward. The reward is a predefined function giving a concrete measure of the method's performance. By learning this way, RL methods do not require accurate models of the components in the control loop and, hence, can be viewed as an automated approach for control.

Previous work in RL-based adaptive optics control has focused on either controlling DM modes using model-free methods that learn a policy $\pi_\theta : s_t \mapsto a_t$ parameterized by θ that maps states s_t (or observations) into actions a_t directly (Landman et al. 2020, 2021; Pou et al. 2022), or using model-based methods that employ a planning step to compute actions (Nousiainen et al. 2021). The model-free methods have the advantage of being fast to evaluate, as the learned policies are often neural networks that support sub millisecond inference. However, they suffer from the large space of actions resulting from the number of actuators that need to be controlled in adaptive optics systems – learning to control each actuator simultaneously with a model-free method is difficult. On the other hand, model-based RL approaches benefit from being simple to train using even off-policy data, that is, data obtained, while using a different (e.g., classical integrator) control method. A Model-based method may only need hundreds of iterations while a model-free algorithm such as policy gradient methods may need millions of iterations (Janner et al. 2019). However, the planning step of model-based RL is often iterative and could, therefore, be too slow for AO control, even with expensive hardware (Nousiainen et al. 2021).

In this paper, we unify the approaches described above by learning a dynamics model and using the model to train a policy that is fast to evaluate and scales to control all actuators in a system. We call this hybrid algorithm Policy Optimization for Adaptive Optics (PO4AO). We do this by employing an end-to-end convolutional architecture for the policy, leveraging the differentiable nature of the chosen reward function, and directly backpropagating through trajectories sampled from the model. Our method scales to sub-millisecond inference, and we present promising results in both a large pyramid-sensor-based simulation and a laboratory setup using Magellan Adaptive Optics eXtreme system (MagAO-X, Males et al. 2018), where our method is trained from scratch using interaction.

2. Related work

The adaptive optics control problem differs from the typical control problems considered by modern RL research. The main challenges of AO control are two-fold: first, the control space is substantially larger than in classical RL literature and is typically parameterized by 500–10 000 degrees of freedom (DoF). Secondly, the state of the system is observed through an indirect measurement, where the related inverse problem is not well-posed. On the bright side, it has been observed in the literature that simple differentiable reward functions with a relatively short time horizon can lead to good performance (Nousiainen et al. 2021).

Recently, progress has been made toward full reinforcement learning-based adaptive optics control. Landman et al. (2020) use the model-free recurrent deterministic policy gradient algorithm to control the tip and tilt modes of a DM and a variation of the method to control a high order mirror in the special case of ideal wavefront sensing. Pou et al. (2022) implemented a model-free multi-agent approach to control a 40 × 40 Shack-Harmann-based AO system and analyzed the robustness against noise and variable atmospheric conditions. On the other hand, Nousiainen et al. (2021) present a model-based solution that learns a dynamics model of the environment and uses it with a planning algorithm to decide the control voltages at each timestep. This method shows good performance but requires heavy computation at each control loop iteration, which will be a problem in future generations of instruments with more actuators per DM. PO4AO aims for the best of both worlds: it requires only a small amount of training data and has a high inference speed, capable of scaling to modern telescopes. Further, we analyze the performance of our method in different noise levels and varied wind conditions combined with nonlinear wavefront sensing.

In RL terms, model-based policy optimization is an active area of research. Work that tackles the full reinforcement learning problem without assuming a known reward function includes Heess et al. (2015), and Janner et al. (2019). In contrast, PILCO and the subsequent deep PILCO (Deisenroth & Rasmussen 2011; Gal et al. 2016) are methods that directly backpropagate through rewards. Our method is similar to deep PILCO in the sense that it learns a neural network policy from trajectories sampled from a neural network dynamics model.

In addition, significant progress has also been made in AO control methods outside RL and fully data-driven algorithms. Linear-quadratic-Gaussian control (LQG) based methods have been studied in Kulcsár et al. (2006); Paschall & Anderson (1993); Gray & Le Roux (2012); Conan et al. (2011); Correia et al. (2010a,b, 2017), sometimes combined with machine learning for system identification (Sinquin et al. 2020). Predictive controllers have been studied in Guyon & Males (2017); Poyneer et al. (2007); Dessenne et al. (1998); van Kooten et al. (2017, 2019). Methods vary from linear filters to filters operating on single modes (such as Fourier modes) to neural network approaches (Swanson et al. 2018; Sun et al. 2017; Liu et al. 2019; Wong et al. 2021). Predictive control methods have also been studied in a closed-loop configuration. Males & Guyon (2018) address a closed-loop predictive control's impact on the postcoronagraphic contrast with a semianalytic framework. Swanson et al. (2021) studied closed-loop predictive control with NNs via supervised learning, where a NN is learned to compensate for the temporal error.

Finally, other RL-based methods have been developed for different types of AO. In order to mitigate alignment errors in calibration, a deep-learning control model was proposed in Xu et al. (2019). A model-free RL method for wavefront sensorless AO was studied in Ke et al. (2019). The method is shown to provide faster corrections speed than a baseline method assuming a relatively low-order AO system, while our work focuses on the case of XAO for HCI.

3. Reinforcement learning applied to adaptive optics

Since we introduce a novel approach (RL) to the field of AO, we present hereafter some of the standard notations and terms used in RL. The de facto mathematical framework for modeling sequential decision problems in the field of RL is the "Markov Decision Process" (MDP). An MDP is a discrete-time stochastic process which, at time step t, is in a "state" $\boldsymbol{s}_t \in \mathcal{S}$ where $\mathcal{S}$ is the set of all possible states. A "decision-maker" then takes an "action" $\boldsymbol{a}_t \in \mathcal{A}$ (again, $\mathcal{A}$ is the set of possible actions) based on the current state, and the "environment" changes to the next state $\boldsymbol{s}_{t+1}$. As the transition dynamics $(\boldsymbol{a}_t, \boldsymbol{s}_t) \mapsto \boldsymbol{s}_{t+1}$ is random in nature (influenced e.g. by the turbulence evolution) it is represented here by the conditional probability density function $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$[2]. At each timestep a "reward" $R_t = r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is also observed, which is a (possibly stochastic) function of the current state and action. The modeler usually designs the reward to make the decision-maker produce some favorable behavior (e.g., correcting for turbulence distortions).

The actions our decision-maker takes are determined by a "policy" $\pi_\theta : \boldsymbol{s}_t \mapsto \boldsymbol{a}_t$, which is a function that maps states into actions. For example, the matrix-vector multiplier (MVM) can be viewed as a policy, taking a wavefront sensor measurement as input and outputting the control voltages. The objective of reinforcement learning is to find a policy such that

$$\arg\max_{\theta} \mathbb{E}_{p_\theta(\mathbf{s}_0,\ldots,\mathbf{s}_T)} \left[\sum_{t=0}^{T} r(\boldsymbol{s}_t, \pi_\theta(\boldsymbol{s}_t)) \right], \tag{1}$$

where

$$p_\theta(\mathbf{s}_0, ..., \mathbf{s}_T) = p_0(\boldsymbol{s}_0) \prod_{t=1}^{T} p(\boldsymbol{s}_t|\mathbf{s}_{t-1}, \pi_\theta(\mathbf{s}_{t-1}))$$

with the initial distribution $\boldsymbol{s}_0 \sim p_0$ and convention $\pi_\theta(\mathbf{s}_{-1}) = \boldsymbol{a}_0$ for a fixed initial DM commands $\boldsymbol{a}_0$. In particular, we focus here on parametric models of π_θ where θ is the set of parameters of the policy, for example, the weights and biases of a neural network. That is, given that the actions are given by π_θ, we wish to find the parameters θ that maximize the expected cumulative reward the decision-maker receives. Here T is the maximum length of an episode or a single run of the algorithm in the environment.

The transition dynamics is usually not known in adaptive optics control: it includes a multitude of unknowns including the atmosphere turbulence, dynamics of the WFS and DM, and the jitter in the computational delay. In order to solve Eq. (1) efficiently, model-based RL algorithms estimate the true dynamics model $p(s_{t+1}|s_t, a_t)$ in Eq. (1) by an approximate model $\hat{p}(s_{t+1}|s_t, a_t)$. Model-free methods, in turn, only learn a policy – they do not attempt to model the environment.

The standard MDP formulation assumes that all information about the environment is contained in the state s_t. This is not the case in many real-world domains, such as adaptive optics control. A more refined formulation is then the "partially observed" MDP or POMDP, where the decision-maker observes o_t, which is some subset or function of the true underlying state. The Markov property, that is, the assumption that the next state depends only on the previous state and action, does not necessarily apply to the observations in a POMDP. This work uses the standard method of having our state representation include a small number of past observations (WFS measurements) and actions (control voltages) to deal with this issue. This allows the policy to use knowledge of past actions to predict the next action. The exact form of the observations o_t and the full state s_t for adaptive optics control will be given in Sect. 5.1.

Finally, it is common in RL to use reward functions that are not differentiable (such as 1 for winning a game, 0 otherwise) or functions that do not depend directly on the state. In high-contrast imaging, we would like to minimize the speckle intensity in the post-coronagraphic PSF. However, this can be difficult to estimate at the high frequencies of modern HCI instruments. We discuss the specific choices in this regard in Sect. 5.1.

4. Adaptive optics control

This section introduces AO control aspects that are relevant to our work. First, we introduce the AO system components and then outline a standard control law called the integrator and the related calibration process. An overview of the AO control loop is given in Fig. 1; the incoming light ϕ_t^{tur} at the timestep t gets corrected by the DM. Next, the WFS measures the DM corrected residual wavefront ϕ_t^{res}. After receiving the wavefront sensor measurement, the control computer calculates a set of control voltages and sends the commands to the DM.

Further, the AO control loop inherits a temporal delay. The delay consists of a measurement delay introduced by the WFS integration and a control delay consisting of WFS readout, computation of the correction signal by the control algorithm, and its application to the DM. These add up to a total delay of at least twice the operating frame-time of the AO system (Madec 1999).

[2] The initial state is drawn from the initial state distribution $\boldsymbol{s}_0 \sim p_0(\boldsymbol{s}_0)$.

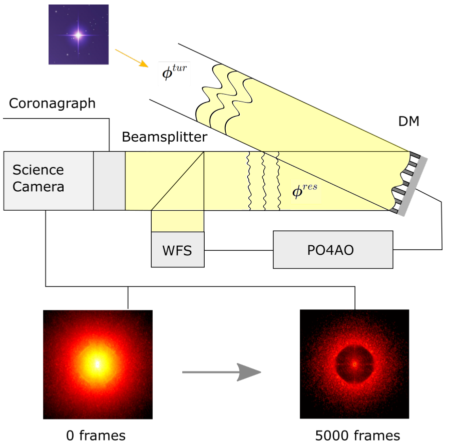


Fig. 1. Overview of the AO control loop and the performance of PO4AO. The method, PO4AO, feeds actions to the environment, observes the outcome, and then improves the control regarding the reward. Starting from a random behavior at first (frame 0), the method learns a predictive control strategy in only 5000 frames of interaction.

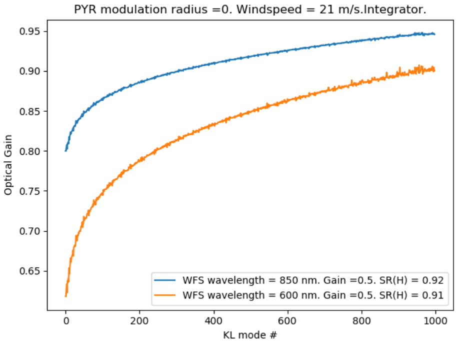


Fig. 2. Modal optical gains for the case of an 8-m telescope with zero-modulation and integrator control and considering two different wavefront sensor wavelengths.

4.1. Pyramid wavefront sensor for adaptive optics

The function of the WFS is to measure the spatial shape of the residual phase of a wavefront ϕ_t^{res}. There are several different types of WFSs, but in this work, we focus on the so-called pyramid WFS (PWFS), which is a mature concept providing excellent performance for HCI (Guyon 2005). In the following, we give a short description of the PWFS.

The PWFS can be viewed as a generalization of the Foucault knife-edge test (Ragazzoni 1996). In pyramid wavefront sensing, the electric field of the incoming wavefront is directed to a transparent four-sided pyramid prism. The prism is located in the focal plane of an optical system and, hence, can be modeled as a spatial Fourier filter that introduces specific phase changes according to the shape of the prism (Fauvarque et al. 2017). This four-sided pyramid divides the incoming light into four different directions, and most of the light is propagated to four intensity images on the PWFS detector. Due to the slightly different optical paths of the light, the intensity fields differ from each other. These differences are then used as the data for recovering the disturbances in the incoming phase screen.

Commonly, pyramid data, that is, the intensity fields, are processed to so-called slopes w_x, w_y that correlate positively to actual gradients fields of the phase screen. In this paper, we follow the approach of Vérinaud (2004), where the slopes are normalized with the global intensity. In practice, we receive a vector $\boldsymbol{w}$ that is a collection of the measurements w_x, w_y at all possible locations x, y.

Both modulated and nonmodulated pyramid sensor observations are connected to the incoming wavefront via a nonlinear mathematical model. This study considers nonmodulated PWFSs, where the nonlinearity is stronger, but the sensitivity is better at all spatial frequencies (Guyon 2005). Currently, most wavefront reconstruction algorithms utilize a linearization of this model, inducing a trade-off between sensitivity and robustness (modulated PWFS vs. nonmodulated PWFS). Machine learning techniques have the potential to overcome this trade-off and increase PWFS performance without a decisive robustness penalty.

Another feature of the PWFS is that its sensitivity varies depending on both the seeing conditions and the level of AO correction itself (Korkiakoski et al. 2008) which is mainly introduced by high spatial frequency aberrations which the DM cannot control. The presence of these aberrations reduces the signal strength of the measurement also for the controlled modes, and the strength of the reduction depends on the mode's spatial frequencies (Korkiakoski et al. 2008).

To illustrate the OG effect of the Pyramid sensor, we use a preliminary version of a semi-analytical model code-named "AO cockpyt" (in prep.). This model is based on the work of Fauvarque et al. (2019), describing the sensitivity of the Pyramid sensor in the presence of residuals, and on an adaptation of Fourier models from Jolissaint (2010) and Correia et al. (2020). Figure 2 shows the analytically predicted modal optical gains for the case of an 8-m telescope with zero-modulation and integrator control and considering two different wavefront sensor wavelengths. The assumed AO system for this analytical prediction is the same as the one used for our numerical simulations presented in Sect. 6 (41 × 41 actuators correct for seeing of 0.7″ at 550 nm at 1000 Hz framerate using a 0th magnitude guide star). The figure shows how the optical gain depends on the spatial frequency of the control modes (the K-L are numbered from low- to high spatial frequencies) and on the WFS Strehl ratio, which is lower at the shorter wavelength.

A modal optimization of the controller gains using the knowledge of Fig. 2 can solve most of the problems (diagonality assumption in Chambouleyron et al. 2020) and applying the usual control theory margins (gain and phase) for ensuring a robust system. Determining optical gains in real-time is possible but complex (Deo et al. 2021; Chambouleyron et al. 2020), and the relative variations shown in Fig. 2 are of the order 10–20% for our XAO case. Hence, compensation for the mode-dependent optical gains with a single integrator gain may lead to acceptable results. However, an aggressive static integrator gain could impair loop robustness when the correction improves, and the optical gains increase. Section 6 presents evidence that PO4AO takes the PWFS OG effect into account for improved performance. Further, modal gain compensation of OG is a solution that is expected to work in favorable cases, but still, the nonlinearities after OG compensation will remain and can only be treated with nonlinear methods as the one studied in this paper.

4.2. Classical adaptive optics control

Classically, an AO system is controlled by combining a linear reconstructor with a proportional-integral (PI) control law. We call this controller the integrator and use it as the reference method for the comparison with PO4AO. As a starting point, the controller assumes to operate in a regime where the dependence between WFS measurements and DM commands is linear to a good approximation, satisfying

$$\boldsymbol{w}_t = Dv_t + \xi_t, \tag{2}$$

where $\boldsymbol{w}_t = (\delta w_t^1, \ldots, \delta w_t^n)$ is the WFS data, v_t the DM commands and D is so-called interaction matrix. Moreover, ξ_t models the measurement noise typically composed of photon and detector noise. The DM command vector v_t represents the DM shape given in the function subspace linearly spanned by the DM influence functions.

The interaction matrix D represents how the WFS sees each DM command. It can be derived mathematically if we accurately know the system components (WFS and DM) and the alignment of the system. In practice, it is usually measured by poking the DM actuators with a small amplitude staying inside the linear range of the WFS, and recording the corresponding WFS measurements (Kasper et al. 2004; Lai et al. 2021).

The interaction matrix D is generally ill-conditioned, and regularization methods must be used to invert it (Engl et al. 1996). Here, we regularize the problem by projecting v_t to a smaller dimensional subspace spanned by Karhunen–Loéve (KL) modal basis. The KL basis is computed via a double diagonalization process, which considers the geometrical and statistical properties of the telescopes (Gendron 1994). This process results in a transformation matrix B_m which maps DM actuator voltages to modal coefficients.

We observe that the modal interaction matrix is now obtained as $DB_m^\dagger$, where $B_m^\dagger$ is the Moore–Penrose pseudo-inverse of B_m. A well-posed reconstruction matrix for the inverse problem in Eq. (2) is then given by

$$C_m = (DP_m)^\dagger, \tag{3}$$

where $P_m = B_m^\dagger B_m$ is a projection map to the KL basis. Regularization by projection is a classical regularization with well-established theory Engl et al. (1996). It is well-suited for the problem at hand due to the physics-motivated basis expansion and fixed finite dimension of the observational data.

With $\Delta\boldsymbol{w}_t$ denoting the residual error seen by the WFS in closed loop, and t denoting the discrete time step of the controller, the integrator control law is

$$\tilde{\boldsymbol{v}}^t = \tilde{\boldsymbol{v}}^{t-1} + gC\Delta\boldsymbol{w}^t, \tag{4}$$

where g is so-called the integrator gain. In literature, $g < 0.5$ is typically found to provide stable control for a two-step delay system Madec (1999).

5. Learning to control using a model

Here we detail the control algorithm including optimization for the dynamics model $p_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$ and the policy $\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$. In standard AO terms, the policy combines the reconstruction and control law (e.g., a least-squares modal reconstruction followed by integrator control); in our case, a nonlinear correction to a least-squares modal reconstruction (MDP formulation) and a predictive control law. The key idea is to learn a dynamics model that predicts the next wavefront sensor measurement given the previous measurements and actions and to use that model to optimize the policy. Our method iterated the following three phases[3]:

1. Running the policy: we ran the policy in the AO control loop for T timesteps (a single episode).
2. Improving the dynamics model: we optimized the dynamics model using a supervised learning objective Eq. (9).
3. Improving the policy: we optimized the policy using the dynamics model Eq. (12).

At each iteration of our algorithm, we collected an episode's worth of data, e.g., 500 subsequent sensor measurements and mirror commands, by running the policy in the AO control loop for T timesteps. We then saved the observed data and given actions and trained our policy and dynamics model using gradients computed from all previously observed data.

The following sections discuss how we represented each observation, our convolutional neural network architecture for both the dynamics model and the policy, and the optimization algorithm itself.

5.1. Adaptive optics as a Markov decision process

We defined the adaptive optics control problem as an MDP by following the approach of Nousiainen et al. (2021). As discussed in Sects. 3 and 4, we do not directly observe the state of the system but instead observe a noisy WFS measurement. In addition, adaptive optics systems suffer from control delay resulting from the high speed of operation, which means that the system evolves before the latest action has been fully executed. Hence, we set our state presentation to include a small amount of past WFS measurements and control voltages.

We denote the control voltages applied to DM at a given time instance t by $\tilde{\boldsymbol{v}}_t$ and the preprocessed PWFS measurements by $\boldsymbol{w}_t$. We defined the set of actions to be the set of differential control voltages:

$$\boldsymbol{a}_t = \Delta\tilde{\boldsymbol{v}}_t. \tag{5}$$

In adaptive optics, at each timestep t, we observe the wavefront sensor measurement $\boldsymbol{w}_t$. We project the measurement into voltage space by utilizing the reconstruction matrix C. The observation is then given by the quantity:

$$\boldsymbol{o}_t = C\boldsymbol{w}_t. \tag{6}$$

To represent each state, we concatenated previous observations and actions. That is,

$$\boldsymbol{s}_t = \left(\boldsymbol{o}_t, \boldsymbol{o}_{t-1}, \ldots, \boldsymbol{o}_{t-k}, \boldsymbol{a}_{t-1}, \boldsymbol{a}_{t-2}, \ldots, \boldsymbol{a}_{t-m}\right), \tag{7}$$

where we chose $k = m$ (as in the typical pseudo-open-loop prediction). The state includes data from the previous m time steps and the reconstruction matrix C. Here the reconstruction matrix serves solely as a preprocessing step for WFS measurements. It speeds up the learning process by simplifying the convolutional NN (CNN) architecture (same dimensional observations and actions). However, It does not directly connect the measurement to actions and, therefore, using it does not imply a sensitivity to misregistration (Nousiainen et al. 2021).

[3] See Algorithm 1 for more details.

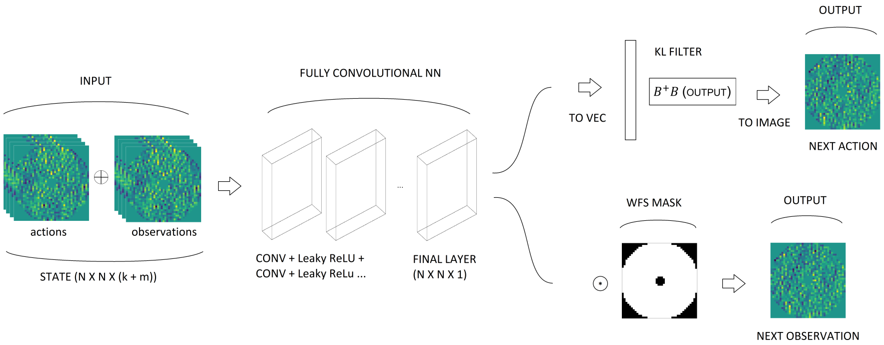


Fig. 3. Neural network architectures. Both the dynamics model and the policy NN take same input: concatenations of past actions and observations. They also share the same fully convolutional structure in the first layers. At the output layer, the policy model includes the KL-filtering scheme (*upper right corner*) and the dynamics model output is multiplied with the WFS mask (*lower right corner*). See Sect. 5.2 for details.

For a state-action pair, the reward was chosen as the residual voltages' negative squared norm corresponding to the following measurement:

$$r(\boldsymbol{s}_t, \boldsymbol{a}_t) = -\mathbb{E}_{p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)} ||\tilde{\boldsymbol{o}}_{t+1}||^2, \tag{8}$$

where $\tilde{\boldsymbol{o}}_{t+1}$ was obtained from $\tilde{\boldsymbol{s}}_{t+1} \sim p(\cdot|s_t, a_t)$.

This quantity is proportional to the observable part of the negative norm of the true residual wavefront. This reward function does not capture all error terms such as aliasing and non-common path errors (NCPA), and hence, the final contrast performance will always be limited by these. The aliasing errors could be mitigated with traditional means, e.g., by introducing a spatial filter (Poyneer & Macintosh 2004) or by oversampling the wavefront, that is, by using a WFS with finer sampling than the one provided by the DM. We also already eluded on the fact that minimizing the residual wavefront seen by the WFS does not necessarily minimize the residual halon in the science image because of NCPA between the two. The PO4AO could treat NCPA by including science camera images in the state formulation, but these would have to be provided at the same cadence as the WFS data, which is usually not the case. Still, NCPA can be handled by PO4AO in the usual way by offsetting the WFS measurements by an amount determined by an auxiliary image processing algorithm (e.g., Give'on et al. 2007; Paul et al. 2013). Finally, the reward does not include an assumption on the time delay of the system, so the method learns to compensate for any delay and predict the wavefront.

5.2. The dynamics model

An adaptive optics system inherits strong spatial correlations in observations and control space – neighboring actuators and WFS pixels close to each are more correlated than actuators further apart due to the steep negative slope of the turbulence temporal PSD (Fried 1990) and the frozen flow hypothesis. We employed a standard fully convolutional neural network (CNN), equipped with a leaky rectified linear unit (LReLU, Maas et al. 2013) activation functions that predicts the next wavefront sensor readout. The CNN should work well for our setup with DM actuators and WFS subapertures aligned on a grid in a spatially homogeneous geometry.

In practice, the state is a 3D tensor (matrices stack along the third dimension, that is, a $(N \times N \times (k + m))$ tensor) with the channel dimension corresponding to DM actuator grid (2D) and the number of previous observations (k) and actions (m). See Fig. 3 for an illustration.

The deterministic dynamics model $\hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$ estimates the next state $\boldsymbol{s}_{t+1}$ given the previous state and action. The model parameters ω (i.e., the NN weights and biases) were trained by first running the policy π in the environment, that is, controlling the AO system with the policy, collecting tuples of $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ into a dataset $\mathcal{D}$, and minimizing the squared difference between the true next states and the predictions

$$\sum_{\mathcal{D}} ||\boldsymbol{s}_{t+1} - \hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)||^2 = \sum_{\mathcal{D}} ||\boldsymbol{o}_{t+1} - \hat{\boldsymbol{o}}_{t+1}||^2, \tag{9}$$

where $\boldsymbol{o}_{t+1}$ is obtained from the state $\boldsymbol{s}_{t+1}$ and $\hat{\boldsymbol{o}}_{t+1}$ is the observation predicted by $\hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The optimization was done using the Adam algorithm (Kingma & Ba 2014). Again we did not assume any integer time delay here, but as the past actions are included in the state formulation, we learned to compensate for it.

It is well-known that model-based RL performance unfavorably exploits an overfitted dynamics model in the control (e.g., planning or policy optimization), especially in the early stages of training (Nagabandi et al. 2018). To discourage this behavior, we employed an ensemble of several models, each of which is trained using different bootstrap datasets, that is, subsets of the observations collected during training. In practice, this means that each model sees a different subset of observations, leading to different NN approximations. During policy training, predictions are averaged over the models (line 9 of Algorithm 1). See, for example, Chua et al. (2018) for a more detailed discussion on ensemble models.

5.3. The policy model

Again, we employed a fully convolutional neural network as the policy, similar to the dynamics model. The input is a 3D tensor representing the state, and the output a 2D tensor (a matrix) representing the actuator voltages. The WFS measurement is blind or insensitive to some shapes of the mirror, such as the well-known waffle mode and actuators on the boundary. We ensured that we do not control these modes by projecting each set of control voltages to the control space, that is, we reshaped the 2D output to a vector, multiplied it by a filter matrix, and then reshaped the output back to a 2D image. The full policy model π is given by

$$\pi_\theta(\boldsymbol{s}_t) = B^\dagger B F_\theta(\boldsymbol{s}_t), \tag{10}$$

where $B^\dagger B$ projects the control voltages onto the control space defined by the K-L modes and F_θ is the standard fully convolutional NN, where the output is vectorized. Figure 3 gives more detailed overview of the network architecture of F_θ.

5.4. Policy optimization

Ideally, the policy $\pi_\theta(\boldsymbol{s}_t)$ would be optimized based on the expected cumulative reward function Eq. (1). However, as we do not have access to the true dynamics model p, we must approximate it with the learned dynamics model $\hat{p}_\omega$. To stabilize this process, we introduced an extended time horizon $H \ll T$ over which the performance was optimized. Let us define

$$\hat{r}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t) = -\|\tilde{\boldsymbol{o}}_{t+1}\|^2, \tag{11}$$

where $\tilde{\boldsymbol{o}}_{t+1}$ is obtained from $\tilde{\boldsymbol{s}}_{t+1} = \hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$. This leads to the approximative policy optimization problem

$$\arg\max_\theta \sum_{\mathbf{s}\in\mathcal{D}} \sum_{t=1}^{H} \hat{r}_\omega(\tilde{\boldsymbol{s}}_t, \pi_\theta(\tilde{\boldsymbol{s}}_t)), \tag{12}$$

where H the planning horizon and

$$\tilde{\mathbf{s}}_1 = \mathbf{s} \quad \text{and} \quad \tilde{\mathbf{s}}_{t+1} = \hat{p}_\omega(\tilde{\boldsymbol{s}}_t, \pi_\theta(\tilde{\boldsymbol{s}}_t)).$$

Here the planning horizon H was chosen based on the properties of the AO system. More precisely, for AO control, the choice of the planning horizon H is driven by the system's control delay. In the case of a simple two-frame delay, no DM dynamic, and no noise, we would plan to minimize the observed wavefront sensor measurements two steps into the future, that is, we would implicitly predict the best control action by the DM at the time of the corresponding WFS measurement. However, the effective planning horizon is longer in the presence of DM dynamics and temporal jitter since the control voltage decisions are not entirely independent. The choice of the planning horizon is a compromise between two effects: too short a planning horizon jeopardizes the loop stability, and too long a planning horizon makes the method prone to overfitting. We used $H = 4$ frames in all our experiments (numerical and laboratory) as a reasonably well-working compromise.

The policy π was optimized by sampling initial states from previously observed samples, computing actions for them, and using the dynamics model to simulate what would happen if we were to take those actions. We could then use the differentiable nature of both our models and the reward function to backpropagate through rewards computed at each timestep. More specifically, at each iteration, we sampled a batch of initial states s_τ and computed the following H states using the dynamics model. We then had H rewards for each initial state, and we used the gradients of the sum of those rewards with respect to the policy parameters θ to improve the parameters. The full procedure of training the dynamics and the policy is given in Algorithm 1, where the while-loop (line 3) iterates over episodes and lines 6–16 execute an update of policy via policy optimization.

Algorithm 1 Policy Optimization for Adaptive Optics (PO4AO)

```
 1: Initialize policy and dynamics model parameters $\theta$ and $\omega$ randomly
 2: Initialize gradient iteration length $K$, batch size $B < |\mathcal{D}|$ and planning horizon $H$
 3: while not converged do
 4:     Generate samples $\{s_{t+1}, s_t, a_t\}$ by running policy $\pi_\theta(a_t|s_t)$ for $T$ timesteps (an episode) and append to $\mathcal{D}$
 5:     Fit dynamics by minimizing Eq. (9) w.r.t $\omega$ using Adam
 6:     for iteration $k = 1$ to $K$ do
 7:         Sample a mini batch of $B < |\mathcal{D}|$ states $\{s_\tau\}$ from $\mathcal{D}$
 8:         for each $\mathbf{s}_\tau$ in the mini batch do
 9:             Set $\tilde{\mathbf{s}}_1^\tau = \mathbf{s}_\tau$
10:             for $t = 1$ to $H$ do
11:                 Predict $\boldsymbol{a}_t = \pi_\theta(\boldsymbol{s}_t)$
12:                 Predict $\boldsymbol{s}_{t+1} = \hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$
13:                 Calculate $R_t = \hat{r}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$
14:             end for
15:         end for
16:         Update $\theta$ by taking a gradient step according to $\nabla_\theta \sum_{t=\tau}^{\tau+H} R_t$ with Adam.
17:     end for
18: end while
```

6. Numerical simulations

6.1. Setup description

We evaluate the performance of PO4AO by numerical simulations. We used the COMPASS package (Ferreira et al. 2018) to simulate an XAO system at an 8-m employing a nonmodulated Pyramid WFS in low noise (0 mag) and moderately large noise (9 mag) conditions. For comparison, we also considered the theoretical case of an "ideal" wavefront sensor where the wavefront reconstruction is simply a projection of the 2D-turbulence screen onto the DM's influence functions.

We also include a simulation of a 40-meter telescope XAO with PWFS to confirm that PO4AO nicely scales with aperture size and XAO degrees of freedom. Comprehensive error analysis and fine-tuning are left for future work. In order to stabilize the performance of the integrator, we added $2\lambda/D$ modulations to the PWFS.

For all simulations, we simulated the Atmospheric turbulence as a sum of three frozen flow layers with Von Karman power spectra combining for Fried parameter $r0$ of 16 cm at 500 nm wavelength. The complete set of simulation parameters is provided in Table 1.

We compare PO4AO against a well-tuned integrator and instantaneous controller, not affected by measurement noise or temporal error. For the Pyramid WFS, it still propagates aliasing and the fitting error introduced by uncontrolled or high-spatial frequency modes. For the idealized WFS, it acts as a spatial high-pass filter, instantaneously subtracting the turbulent phase projected on the DM control space (DM fitting error only).

In particular, we chose the simulation setups to demonstrate the following key properties of the proposed method. Firstly, The method achieves the required real-time control speed while being quick to train. This property enables the controller to be trained just before the science operation and be further updated during the operation. Consequently, the method is trained with the most relevant data and does not need to generalize to all possible conditions at once; furthermore, the method retains these properties with an ELT-scale instrument. Secondly, The method is a

Table 1. Simulations parameters.

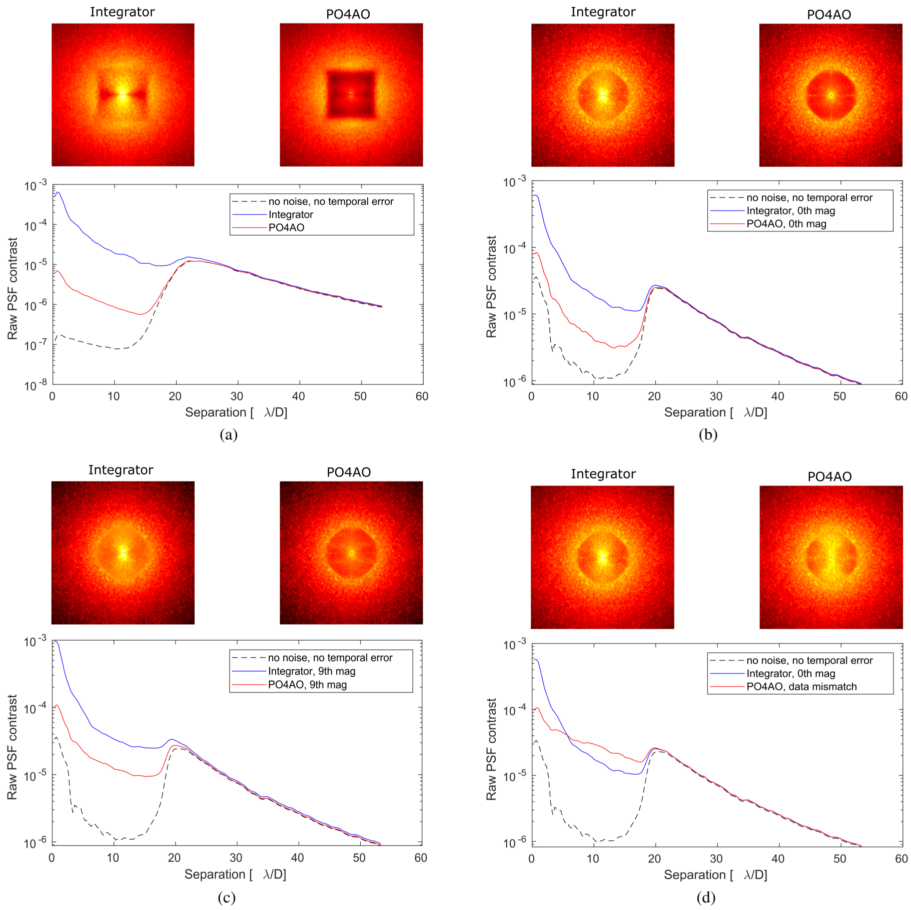
Parameter	Value	Units
Telescope "VLT"		
Telescope diameter	8	m
Obstruction ratio	14	percent
Sampling frequency	1000	Hz
Active actuators	1364	actuators
PWFS subapertures	41 × 41	apertures
PWFS modulation	0	λ/D
Photon flux 0/9 mag	$1.25 \times 10^8/3.1 \times 10^4$	photons/frame/aperture
DM coupling	0.3	percent
DM influence functions	"Gaussian"	···
WFS wavelength	0.85	µm
Science camera wavelength	1.65	µm
Telescope "ELT"		
Telescope diameter	40	m
Obstruction ratio	30	percent
Sampling frequency	1000	Hz
Active actuators	10 556	actuators
PWFS subapertures	121 × 121	apertures
PWFS modulation	2	λ/D
Photon flux 0th mag	2.7×10^9	photons/frame/aperture
DM coupling	0.3	percent
DM influence functions	"Gaussian"	···
WFS wavelength	0.85	µm
Science camera wavelength	1.65	µm
Atmosphere parameters		
Fried parameter	16	cm at 500 nm
Number of layers	3	···
Layer altitudes	0/4/10	km
C_N^2	50/35/15	percent (%)
Wind speeds	10/26/35	m/s
Wind directions	0/45/180	degrees
L_0 (m)	30/30/30	m
PO4AO parameters		
Planning horizon (H)	4	steps
Past DM commands (m)	15	commands
Past WFS measurements (k)	15	frames
CNN ensemble size	5	···
Dynamics iterations/episode	15	steps
Policy iterations/episode	10	steps
Training mini batch size	32	···
Fixed CNN parameters		
Number of conv. layers	3	layers
Filter size	3 × 3	pixels
Padding	1	pixels
Activation functions	Leaky ReLU	···

predictive controller, robust to nonlinear wavefront sensing and photon noise. Thirdly, The method can cope with the optical gain effect of the pyramid sensor.

6.2. Algorithm setup

We chose the state $\boldsymbol{s}_t$ (in MDP) to consist of 15 latest observations and actions and set the CNN (dynamics and policy) to have 3-layers with 32 filters each. For further details on these choices, see Sect. 6.3. The episode length was set to 500 frames.

Each simulation started with the calibrations of the system and the deriving of the reconstruction matrix C and the K-L basis B; see Sect. 4. We note that the reconstruction matrix C serves solely as a filter that projects WFS measurement to control space. It does not have to match the actual registration of DM and WFS (Nousiainen et al. 2021). In particular, the reconstruction matrix is measured around the null point in the calibrations and, hence,



(a) (b) (c) (d)

Fig. 4. Raw PSF contrast in VLT-scale telescope experiments. *Upper images*: raw PSF contrast. *Lower plot*: the radial averages over the image. The blue lines are for the integrator and red for the PO4AO. The raw PSF contrast was computed during the 1000 frames of the experiment. *Panel a*: performance of PO4AO with ideal WFS. We see that P04AO delivers a factor of 20–90 improvement inside the AO control radius compared to well-tuned integrator. *Panel b*: performance of PO4AO on 0th mag guide star and a nonmodulated PWFS. PO4AO delivers a factor of 4–7 better contrast inside the AO control radius. *Panel c*: performance of PO4AO on 9th mag guide star. We see a factor of 3–9 improvement in the raw PSF contrast. *Panel d*: performance of PO4AO under heavy data mismatch. PO4AO was trained with drastically different wind conditions. The PO4AO still delivers better contrast with small angular separations.

it suffers from the optical gain effect Korkiakoski et al. (2008). For PWFS simulations, the K-L filter was set to include 85% of total degrees of freedom, and for ideal wavefront sensing to filter matrix was an identity, that is, no filtering included.

For all different conditions and instruments, we let simulations run until the performance of PO4AO is converged. That is 46 000 frames (46 s in real-time (theoretical)) with an episode length of 500 frames. While the final contrast performance shown in Figs. 4b–d and 5 is calculated from the last 1000 frames, we note that the correction performance very quickly passes the integrator performance as shown in Figs. 6a–c, and 7. After each episode, as described in Sect. 5.4, we halted the simulations and updated the dynamics and policy models. Given the shallow convolutional structure (3 – layers and 32 filters per layer) of the NN models and our moderate hardware, the combined (dynamic and policy) training time after each episode was about 1.5 s for VLT (and 7 s for ELT with the same training hyperparameters). For real-time implementation, training the NN models should be completed in the duration of an episode, that is, in 0.5 s (500 frames at 1 kHz). Given that we do not use the latest GPU hardware, and a NN update could also be done at a slower rate than after each episode, it is conceivable that this small gap can be overcome, and a real-time implementation of PO4AO is already possible.

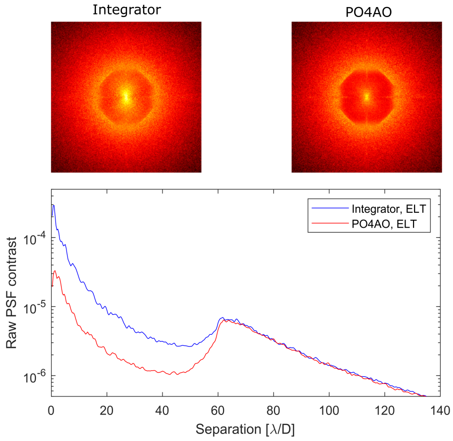


Fig. 5. Raw PSF contrast in ELT-scale experiment.

The dynamics model can also be trained with data obtained with a different controller, such as, the integrator or random control. Therefore, to improve the stability of the learning process, we "warm-up" the policy by running the first ten episodes with the integrator and added binary noise to develop a coarse understanding of the system dynamics:

$$\tilde{\boldsymbol{v}}^t = \tilde{\boldsymbol{v}}^{t-1} + gC\Delta\boldsymbol{w}^t + \sigma x, \tag{13}$$

where x is binary noise (-1 or 1 with the same probability) and $\sigma \in [0, 1]$ is reduced linearly after each episode such that the first episode was run with high binary noise and the 10th episode with zero noise.

6.3. CNN design and MDP state definition

The PO4AO includes two learned models: the policy and the dynamics model. This paper aims to introduce an optimizations method called PO4AO to train the policy (from scratch) that minimizes the expected reward. The algorithm works for all differentiable function classes, for example, neural networks. For simplicity, we chose to model the environment dynamics and policy using generic 3-layer fully convolutional neural networks. While further research is needed in finding the best possible architectures, we experimented with the number of convolutional filters per layer and the number of past telemetry data by testing the algorithm in the "VLT" environment with different combinations; see Table 2. We chose the model CNN 2 to compromise between the overall performance, inference speed for VLT and ELT, and training speed. The chosen model performed well in all simulations and provided fast inference speed and fast training speed such that it could be completed during a single episode. Full model architecture optimization is left for future work (see Sect. 8 for more details).

The inference speed in Table 2 is the speed of the fully convolutional NN architecture inside the policy model (see Fig. 3). The total time control time includes two standard MVMs (preprocessing to voltages + KL filtering in the output layer) in addition to the inference time below. The inference time and training time were run with PyTorch on NVIDIA Quadro RTX 3000 GPU. Note here that given enough parallel computational power (e.g., GPU), the inference time of a fully convolutional NN is more determined by the number of layers and filter (same for VLT and ELT) than the input image's size. We observe that for CNN with fewer filters, the inference speed is very similar for VLT and ELT cases, while for heavier CNNs, the inference speed differs more with the given hardware. The computational time of MVMs is naturally dependent on the DoF.

6.4. Results

6.4.1. Training

To evaluate the training speed of the method, we compare the learning curves (from which 5000 frames are obtained with the integrator + noise controller) of the method to the baseline of the integrator performance under the same realization of turbulence and noise (see Figs. 6b, c, a and 7). Since the simulations are computationally expensive, in the 40-meter telescope experiments, we compare the performance of the PO4AO only to average integrator performance (see Fig. 7).

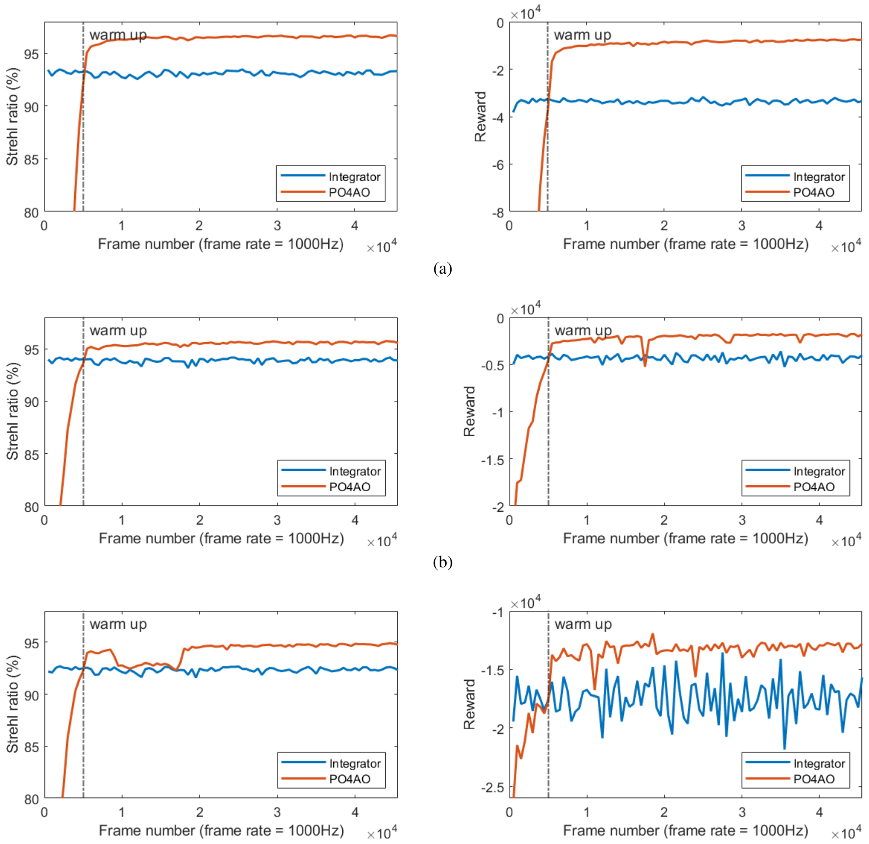
We plotted the training curves with respect to total reward (the sum of normalized residual voltages computed from the WFS measurements) and Strehl ratio side by side. The method tries to maximize the reward, and consequently, it also maximizes the Strehl ratio. In all our simulations, the method achieves better performance than the integrator already after the integrator warm-up of 5000 frames (5 s on a real telescope), and the performance stabilizes at around 30 000 frames (30 s). Since the fully convolutional NN structure can capture and utilize the homogeneous structure of the turbulence, the number of data frames needed for training of VLT and ELT control are on the same scale. However, training the same amount of gradient steps is computationally more expensive (although very parallelizable) for the ELT scale system.

6.4.2. Prediction and noise robustness

Here, we compare the fully converged PO4AO, the integrator, and ideal control in raw PSF contrast. We ran each controller for 1000 frames, and the wavefront residuals for each controller were propagated through a perfect coronagraph (Cavarroc et al. 2006). The raw PSF contrast was calculated as the ratio between the peak intensity of noncoronagraphic PSF and the post-coronagraphic intensity field. A nonpredictive control law suffers from the notorious wind-driven halo (WHD) (Cantalloube et al. 2018), that is, the butterfly-shaped contrast loss in the raw PSF contrast in Figs. 4a–c and 5.

Figure 4a assumes using the ideal WFS, that is, the incoming phase is measured by a noiseless projection of the incoming phase onto the DM. Therefore, the ideal WFS eliminates aliasing and noise in the wavefront reconstruction process, only considering temporal and fitting errors. Further, we can easily eliminate temporal error in a simulation by directly subtracting the measured from the incoming phase. The "no noise, no temporal error" curve (black dashed) in Fig. 4a is therefore only limited by the ability of the DM to fit the incoming wavefront. The integrator with a 2-frame delay (blue curve) is then limited by the temporal error in addition. The PO4AO (red curve) largely reduces the WHD by predicting the temporal evolution of the wavefront but does not fully recover the fitting error limit (black dashed). Figure 4a, therefore, demonstrates the ability of PO4AO to reduce the temporal error.

Figure 4b replaces the ideal WFS with the nonmodulated PWFS, which is affected by aliasing and requires some filtering



(a)

(b)

(c)

Fig. 6. Training plots for 8-meter telescope experiments. *Panel a*: for ideal wavefront sensor, *panel b* is for the 0th magnitude guide star, and *panel c* for the 9th magnitude guide star. The red lines correspond to performance of PO4AO during each episode and blue lines for the integrator. The gray dashed line marks the end of integrator warm up for PO4AO. In all cases the PO4AO outperforms the integrator all ready after the warm up period, in both the Strehl ratio and rewards. An optimized implementation of the PO4AO could run the training in parallel to control, and the training time would then be included in the plot (see Sect. 6.2).

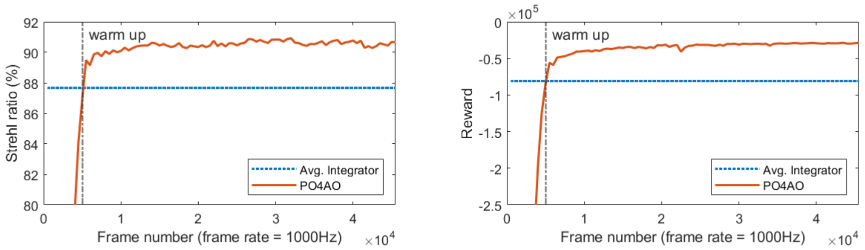


Fig. 7. Training plots for the 40-m telescope experiment. The red lines correspond to performance of the PO4AO during each episode and blue lines for the average integrator performance. The gray dashed line marks the end of integrator warm up for PO4AO. Similarly to 8-meter telescope experiments the PO4AO outperforms the integrator after the warm up.

Table 2. Performance of 11 different 3-layer CNNs.

	CNN design				
	Filters	Past frames (k & m)	Inf. speed (VLT/ELT)	Tr. time/episode (VLT)	Strehl/reward (VLT 0-mag)
CNN 1	32	10	0.29/0.35 ms	1.4 s	95.61/−4101
CNN 2	32	15	0.30/0.37 ms	1.5/7 (ELT) s	95.69/−3340
CNN 3	32	20	0.30/0.40 ms	1.6 s	95.74/−3029
CNN 4	32	25	0.30/0.43 ms	1.8 s	95.75/−2934
CNN 5	64	10	0.30/0.67 ms	2.0 s	95.60/−4002
CNN 8	64	15	0.31/0.70 ms	2.2 s	95.75/−3253
CNN 7	64	20	0.31/0.74 ms	2.5 s	95.75/−3052
CNN 8	64	25	0.32/0.79 ms	2.5 s	95.76/−2845
CNN 9	128	10	0.36/1.52 ms	3.7 s	95.65/−3656
CNN 10	128	15	0.37/1.58 ms	3.8 s	95.71/−2943
CNN 11	128	20	0.38/1.63 ms	4.7 s	95.76/−2847

Notes. All CNN models were trained from scratch with the same PO4AO parameters (see Table 1) and VLT 0-mag simulation environment (see Sect. 6.1 and Table 1). The Strehl and reward were calculated from the last 1000 steps of the experiment. The inference time was also calculated for VLT and ELT-scale systems, while the training time after each episode was only calculated for the VLT-scale system due to computational limitations. The corresponding integrator performance (dominated by the fitting and temporal error) for the "VLT" simulation was 93.59/−10 085 (Strehl/Reward).

of badly seen K-L modes during the reconstruction. Therefore, the "no noise, no temporal error" contrast performance is worse than for the ideal WFS in Fig. 4a. The integrator with a 2-frame delay (blue curve) performs at a very similar contrast as in the ideal WFS case, so it is still limited mostly by temporal error. Again, PO4AO (red curve) lies about halfway between the integrator and "no noise, no temporal error" controllers but performs at a reduced contrast compared to the ideal WFS case. Therefore, the PO4AO performance with the nonmodulated PWS is affected by aliasing and reconstruction errors as well as the temporal error.

Figure 4c adds a significant amount of measurement noise. While this obviously does not affect the "no noise, no temporal error" case, the contrast performance of both integrator and PO4AO is strongly reduced and dominated by noise. Still, PO4AO outperforms the integrator, which demonstrates the resilience of PO4AO against noise-dominated conditions. Finally, Fig. 5 demonstrates that PO4AO maintains its properties in an ELT scale simulation.

Unfortunately, a "black box" controller like PO4AO does not allow us to cleanly separate all individual terms in the error budget because the controller's behavior is to some extent driven by the error terms themselves. However, as discussed above, we explored the relative importance of the individual terms by switching them on and off in our numerical experiments.

6.4.3. Robustness against data mismatch

So far, we have focused on static atmospheric conditions and size of the data set $\mathcal{D}$ is not limited, that is, "ever-growing". However, in reality, the atmospheric conditions are constantly changing, creating a so-called data mismatch problem – the prevailing atmospheric conditions are slightly different from the conditions in which the model was trained. To ensure the method's robustness to data mismatch, we trained the model with very different conditions and then tested the model with the original wind profile by plotting the raw PSF contrast averaged over 1000 frames. We altered the wind by reducing the wind speed by 50 percent and adding 90-degree variations to directions for training, that is, we altered the spatial and temporal statistics of the atmosphere. We do not show the corresponding training plot since it was very similar to Fig. 6b. The result of this experiment is shown in Fig. 4d. The integrator has naturally the same performance as before. The PO4AO still delivers better contrast close to the guide star but suffers from pronounced WDH further from the guide star. Most importantly, the PO4AO is robust and maintains acceptable performance even with heavy data mismatch, which could occur in the unlikely case that atmospheric conditions drastically change from one episode to the next, that is, on a timescale of seconds. Anyhow PO4AO with limited data set size (old data irrelevant data removed) would adapt to such a change and recover the performance within the typical training times discussed in the previous paragraph.

6.5. Sensitivity to the PWFS optical gain effect

The PO4AO uses convolutional NNs and is, therefore, a nonlinear method. Prospects are that it can adapt to nonlinearities in the system, such as the optical gain effect observed for the Pyramid WFS. To examine this property, we run the following experiment. We control the nonmodulated PWFS with PO4AO at 850 and 600 nm and record the policy after training. Then we control the PWFS with the integrator and record, in parallel, the actions PO4AO would have taken. The integrator control results in a correction performance similar to the Strehl ratios derived by the semi-analytical model (Fig. 2). At the shorter wavelength, the PWFS sees larger residuals wavefront errors (in radian) and a stronger effect on the optical gains. However, if the controller can cope with such an effect, which we would expect for PO4AO, the suggested actions should counteract the dampened measurement. In order to validate this, we compare the ratios between the standard deviation of the observations (PWFS measurements) and the standard deviation of suggested PO4AO actions. We define an estimate for the optical gain compensation:

$$e_\lambda \propto \mathrm{std}\left(\boldsymbol{o}^{\lambda}_{\mathrm{int}}\right)/\mathrm{std}\left(\boldsymbol{a}^{\lambda}_{\mathrm{po4ao}}\right), \tag{14}$$

where std is the temporal standard deviation, $\boldsymbol{o}^{\lambda}_{\mathrm{int}}$ the observations while running the integrator, λ the observing wavelength,

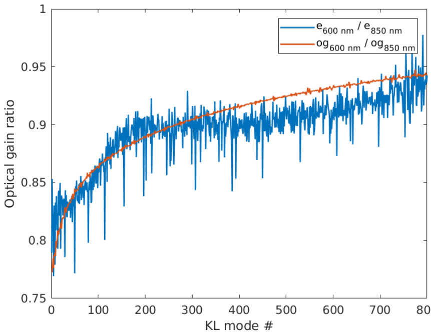


Fig. 8. Sensitivity to the PWFS optical gain effect. The blue line corresponds to ratio between the optical gain estimates between the different wavelengths. The red line is the ratio between the semi-analytically derived optical gains at the two wavelengths (see Sect. 4.1 and Fig. 2).

and $\boldsymbol{a}^{\lambda}_{\mathrm{po4ao}}$ the PO4AO suggested actions. As PO4AO is a predictive control method, this quantity also includes the effect of the prediction, that is, it includes compensation for the temporal error as well. However, we can approximately cancel out the temporal error by comparing the ratio between optical gain estimates obtained at different wavelengths. The result of this experiment is shown in Fig. 8. We see that the empirical estimate for the optical gain sensitivity of PO4AO follows roughly the corresponding ratio of the two semi-analytically derived curves plotted in Fig. 2. In particular, we see that the lower order modes are compensated more than high order modes. We, therefore, conclude that PO4AO adequately compensates for the optical gain effect of the PWFS.

7. Magellan adaptive optics extreme system

In addition to running the numerical simulations presented in the previous section, we also implemented PO4AO on the MagAO-X instrument. MagAO-X is an experimental coronagraphic extreme adaptive optics system that uses woofer-tweeter architecture (ALPAO-97 DM as the woofer and Boston Micromachines 2 K as the tweeter). We use a point source in the f/11 input focus to illuminate the DMs, Pyramid WFS, and scientific camera. Further, we place a classical Lyot coronagraph with a 2.5 λ/D Lyot mask radius in front of the science camera. We set PWFS's modulation ratio to $3\lambda/D$, and the brightness of the guide star is adjusted to match the flux per frame which a 0th magnitude star would provide in 1 ms (i.e., for a system running at 1 kHz.) We used a similar test setup as Haffert et al. (2021b) and ran our experiment by only controlling the woofer DM and injecting disturbances by running simulated phase screens across it. The phase screens were simulated as single-layer frozen flow turbulence with r_0 of 16 cm at 500 nm. We experimented with three different single-layer wind profiles: 5, 15, and 30 m s^{-1}, where the wind speeds correspond to a 1 Khz framerate again.

The PO4AO is implemented with PyTorch and utilizes the Python interface of the MagAO-X RTC to pass data from CPU to GPU memory, do the PO4AO calculations on the GPU, and transfer them back. The data transfer takes time and limits the achievable framerate in this setup to 100 Hz. RTC software that would run entirely on GPUs would not suffer from this limitation.

7.1. The integrator

To retrieve the interactions matrix, we used the standard calibrations process described in Sect. 4. From the interactions matrix, we derived the reconstruction matrix by Tikhonov regularization given by,

$$C = (D^{\top}D + \alpha I)^{-1}D^{\top}, \tag{15}$$

where α is tuned manually. We also tuned the integrator gain manually for each wind profile.

7.2. Policy optimization for adaptive optics

The structure of the MagAO-X experiment is similar to our numerical simulations. First, we trained the PO4AO for 50 episodes (25 000 frames) and then ran for an additional 5000 frames to compare the post-coronagraphic PSFs. We also used the 10 episode warm-up with noisy integrator and the same NN architectures. Given the low number of actuators and the high-order PWFS, we set the number of past telemetry data (k and m) to 10, and instead of filtering 20% of the K-L modes, for maximum performance, we only filter the piston mode in the policy output (see Fig. 3).

7.3. Results

We compare the performance of the PO4AO to the integrator in two ways: by looking at the training curves (see Fig. 9a) and by comparing the post-coronagraphic speckle variance (see Fig. 9c). The PO4AO achieves better performance in all wind conditions than the integrator after 10 k (10 s in theoretical real-time) data frames. The reward is proportional to the mean RMS of the reconstructed wavefront. We further examine the performance by comparing the post-coronagraphic images with the 30 m s^{-1} wind profile; see Figs. 9b,c. The residual intensities of the images (see Fig. 9b) are limited by NCPA. Therefore, instead of comparing the raw PSF contrast, we compare the temporal speckle variance of the method (see Fig. 9c). We see a factor of 3−7 improvement in the speckle variance at 2.4−6λ/D. Given the inner working angle of the coronagraph and DM's control radius, that is where we would also expect to see the improvement. Further, these results are in line with the results from numeric simulations.

8. Discussion

In conclusion, reinforcement learning is a promising approach for AO control that could be implemented in on-sky systems with already existing hardware. The algorithm we propose requires only a small amount of training data and maintains an acceptable performance even when the training conditions differ heavily from test time. Further, it has a high inference speed, capable of scaling to high-order instruments with up to 10k actuators. Thanks to the use of relatively shallow convolutional NN, the inference time is just 300 μs with a modern laptop GPU. The inference time is also similar for an ELT scale system with more than 10k actuators and for a VLT scale system with "just" 1400 actuators.

The method was tested in numerical simulations and a lab setup and provides significantly improved post-coronagraphic contrast for both cases compared to the integrator. It is entirely data-driven, and in addition to predictive control, it can cope with modeling errors such as the optical gain effect and highly nonlinear wavefront sensing. Due to the constantly self-calibrating nature of the algorithm it could turn AO control into a

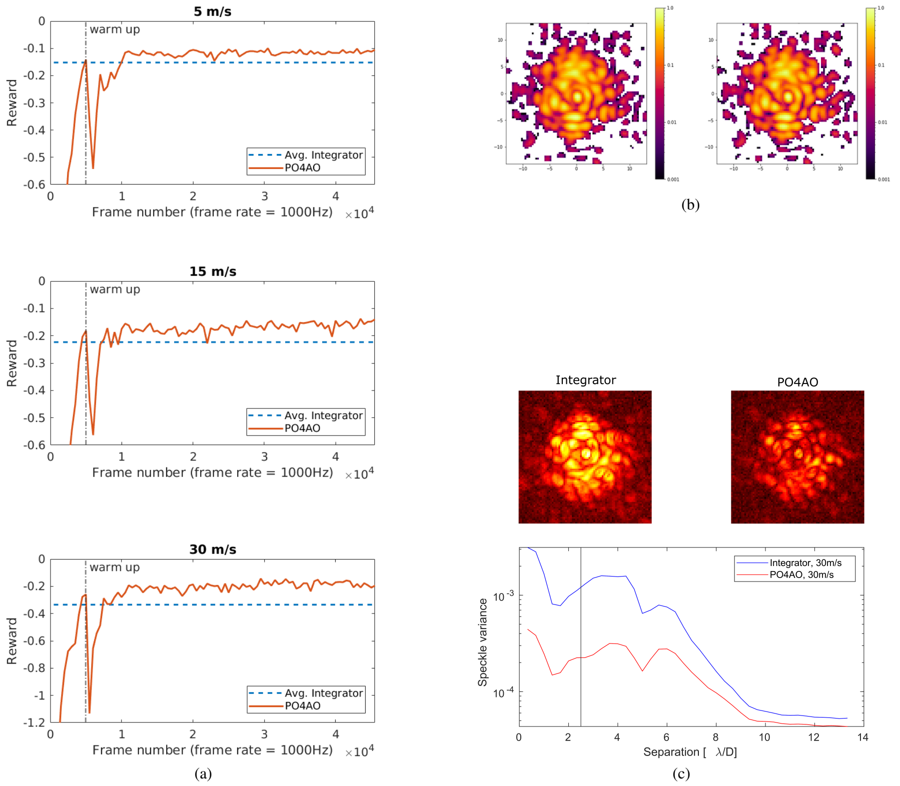


Fig. 9. MagAO-X experiment results. *Panel a*: training curves for PO4AO in the lab setup. The red lines are for PO4AO performance and dashed blue line represents the average integrator performance over an episode. The dashed gray vertical line is where the policy is switch from noisy integrator to PO4AO. For all different wind conditions the PO4AO passes the integrator performance after 10 k frames of data. *Panel b*: MagAO-X post-coronagraphic PSFs of the methods. *Left* is for the Integrator and *right* for the PO4AO. The PSFs are limited by NCPA and, in order to validate the method, we examined the temporal variance of the PSFs (see *panel c*). *Panel c*: temporal variance of MagAO-X post-coronagraphic PSFs. *Upper images*: temporal speckle variance at image plane for both control methods (*left*: integrator, *right*: PO4AO). *Lower image*: radial average over the images. The blue line is for the integrator and the red line for the PO4AO. The gray vertical line represents the inner working angle of the coronagraph (radius $2.5\lambda/D$).

turnkey operation, where the algorithm maintains itself entirely automatically.

We showed that our method is robust to heavy data mismatch, but the performance is reduced for a short time while PO4AO is adapting to the evolution of external conditions. These abrupt changes in wind conditions will rarely occur in the real atmosphere. Therefore, future work should also address maintaining the best possible performance under reasonably varying turbulence. The model learns on a scale of several seconds and can presumably adapt to changing atmospheric conditions at the same time scale. However, more research on the trade-off between model complexity and training speed is still needed. For example, a deeper NN model could generalize better to unseen conditions, while shallower NN models could learn new unseen conditions faster. Currently, the CNN model architectures themselves are not thoroughly optimized, and an exciting research topic would be to find the optimal CNN design to capture the AO control system dynamics for model-based RL. For example, a U-net type CNN architectures (Ronneberger et al. 2015) and mixed-scale dense CNNs (Pelt & Sethian 2018) have shown excellent performance on imaging-related applications. On the other hand, we could utilize similar NN structures that have shown excellent performance in pure predictive control (Swanson et al. 2018, 2021). Such a study should consider a variety of different, preferably realistically changing atmospheric conditions and misalignments as well as prerecorded on-sky data.

As a caveat, the algorithm, like most deep RL methods, is somewhat sensitive to the choice of hyperparameters (e.g., number of layers in neural networks, learning rates, etc.). Moreover, control via deep learning is hard to analyze, and no stability bounds can be established.

Further, development is needed to move from the laboratory to the sky. The method currently runs on a Python interface that has to pass data via the CPU on MagAO-X. To increase the speed of the implementation and the maximum framerates, we must switch to a lower-level implementation that runs both the real-time pipeline and the PO4AO control on the GPU using the same memory banks. In addition, the training procedure needs to run in parallel with the inference, which should be straightforward to implement.

To summarize, this work presents a significant step forward for XAO control with RL. It will allow us to increase the S/N, detect fainter exoplanets, and reduce the time it takes to observe them on ground-based telescopes. As astronomical telescopes become larger and larger, the choice of the AO control method becomes critically important, and data-driven solutions are a promising direction in this line of work. Deep learning and RL methods are transforming many fields, such as protein folding, inverse problems, and robotics, and there is potential for the same to happen for direct exoplanet imaging.

Acknowledgements. The work of T.H. was supported by the Academy of Finland (decisions 320082 and 326961). Support for the work of S.H. was provided by NASA through the NASA *Hubble* Fellowship grant #HST-HF2-51436.001-A awarded by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Incorporated, under NASA contract NAS5-26555. We are very grateful for support from the NSF MRI Award #1625441 for MagAO-X. Finally, we would like to thank the referee for their insightful comments.

References

Cantalloube, F., Por, E. H., Dohlen, K., et al. 2018, A&A, 620, L10
Cavarroc, C., Boccaletti, A., Baudoz, P., Fusco, T., & Rouan, D. 2006, A&A, 447, 397
Chambouleyron, V., Fauvarque, O., Janin-Potiron, P., et al. 2020, A&A, 644, A6
Chua, K., Calandra, R., McAllister, R., & Levine, S. 2018, in Advances in Neural Information Processing Systems, 4754
Conan, J.-M., Raynaud, H., AR, Kulcsár, C., Meimon, S., & Sivo, G. 2011, in Adaptive Optics for Extremely Large Telescopes (Singapore: World Scientific)
Correia, C., Conan, J.-M., Kulcsár, C., Raynaud, H.-F., & Petit, C. 2010a, in 1st AO4ELT conference-Adaptive Optics for Extremely Large Telescopes, EDP Sciences, 07003
Correia, C., Raynaud, H.-F., Kulcsár, C., & Conan, J.-M. 2010b, J. Opt. Soc. Am. A, 27, 333
Correia, C. M., Bond, C. Z., Sauvage, J.-F., et al. 2017, J. Opt. Soc. Am. A, 34, 1877
Correia, C. M., Fauvarque, O., Bond, C. Z., et al. 2020, MNRAS, 495, 4380
Deisenroth, M., & Rasmussen, C. E. 2011, in Proceedings of the 28th International Conference on machine learning (ICML-11), Citeseer, 465
Deo, V., Gendron, É., Rousset, G., et al. 2019, A&A, 629, A107
Deo, V., Gendron, É., Vidal, F., et al. 2021, A&A, 650, A41
Dessenne, C., Madec, P.-Y., & Rousset, G. 1998, Appl. Opt., 37, 4623
Dressing, C. D., & Charbonneau, D. 2015, ApJ, 807, 45
Engl, H. W., Hanke, M., & Neubauer, A. 1996, Regularization of Inverse Problems (Berlin: Springer Science & Business Media), 375
Fauvarque, O., Neichel, B., Fusco, T., Sauvage, J.-F., & Girault, O. 2017, J. Astron. Teles. Instrum. Syst., 3, 019001
Fauvarque, O., Janin-Potiron, P., Correia, C., et al. 2019, J. Opt. Soc. Am. A, 36, 1241
Fernandes, R. B., Mulders, G. D., Pascucci, I., Mordasini, C., & Emsenhuber, A. 2019, ApJ, 874, 81
Ferreira, F., Gratadour, D., Sevin, A., & Doucet, N. 2018, in 2018 International Conference on High Performance Computing & Simulation (HPCS), IEEE, 180
Fried, D. L. 1990, J. Opt. Soc. Am. A, 7, 1224
Fusco, T., Rousset, G., Sauvage, J.-F., et al. 2006, Opt. Exp., 14, 7515
Gal, Y., McAllister, R., & Rasmussen, C. E. 2016, in Data-Efficient Machine Learning workshop (USA: ICML), 4, 25
Gendron, E. 1994, in European Southern Observatory Conference and Workshop Proceedings, European Southern Observatory Conference and Workshop Proceedings, 48, 187
Give'on, A., Kern, B., Shaklan, S., Moody, D. C., & Pueyo, L. 2007, SPIE, 6691, 66910A
Gray, M., & Le Roux, B. 2012, SPIE, 8447, 84471T
Guyon, O. 2005, ApJ, 629, 592
Guyon, O. 2018, Ann. Rev. Astron. Astrophys., 56, 315
Guyon, O., & Males, J. 2017, AJ, accepted [arXiv:1707.00570]
Haffert, S. Y., Males, J., Close, L., et al. 2021a, SPIE, 11823, 118231C
Haffert, S. Y., Males, J. R., Close, L. M., et al. 2021b, J. Astron. Teles. Instrum. Syst., 7, 029001
Heess, N., Wayne, G., Silver, D., et al. 2015, ArXiv e-prints [arXiv:1510.09142]
Heritier, C., Esposito, S., Fusco, T., et al. 2018, MNRAS, 481, 2829
Janner, M., Fu, J., Zhang, M., & Levine, S. 2019, ArXiv e-prints [arXiv:1906.08253]
Jolissaint, L. 2010, J. Euro. Opt. Soc., 5, 10055
Kasper, M., Fedrigo, E., Looze, D. P., et al. 2004, J. Opt. Soc. Am. A, 21, 1004
Ke, H., Xu, B., Xu, Z., et al. 2019, Optik, 178, 785
Kingma, D. P., & Ba, J. 2014, International Conference for Learning Representations, San Diego, 2015
Korkiakoski, V., Vérinaud, C., & Le Louarn, M. 2008, Appl. Opt., 47, 79
Kulcsár, C., Raynaud, H.-F., Petit, C., Conan, J.-M., & Lesegno, P. V. D. 2006, Opt. Express, 14, 7464
Lagrange, A. M., Gratadour, D., Chauvin, G., et al. 2009, A&A, 493, L21
Lai, O., Chun, M., Dungee, R., Lu, J., & Carbillet, M. 2021, MNRAS, 501, 3443
Landman, R., Haffert, S. Y., Radhakrishnan, V. M., & Keller, C. U. 2020, SPIE, 11448, 1144849
Landman, R., Haffert, S. Y., Radhakrishnan, V. M., & Keller, C. U. 2021, J. Astron. Teles. Instrum. Syst., 7, 039002
Liu, X., Morris, T., & Saunter, C. 2019, in International Conference on Artificial Neural Networks (Berlin: Springer), 537
Maas, A. L., Hannun, A. Y., & Ng, A. Y. 2013, Proc. ICML, 30, 3
Macintosh, B., Graham, J. R., Barman, T., et al. 2015, Science, 350, 64
Madec, P.-Y. 1999, Adaptive Optics in Astronomy (Cambridge: Cambridge University Press), 131
Males, J. R., & Guyon, O. 2018, J. Astron. Teles. Instrum. Syst., 4, 019001
Males, J. R., Close, L. M., Miller, K., et al. 2018, SPIE, 10703, 1070309
Marois, C., Racine, R., Doyon, R., Lafrenière, D., & Nadeau, D. 2004, ApJ, 615, L61
Marois, C., Lafrenière, D., Doyon, R., Macintosh, B., & Nadeau, D. 2006, ApJ, 641, 556
Marois, C., Zuckerman, B., Konopacky, Q. M., Macintosh, B., & Barman, T. 2010, Nature, 468, 1080
Mawet, D., Pueyo, L., Lawson, P., et al. 2012, SPIE Conf. Ser., 8442, 844204
Nagabandi, A., Kahn, G., Fearing, R. S., & Levine, S. 2018, in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 7559
Nousiainen, J., Rajani, C., Kasper, M., & Helin, T. 2021, Opt. Express, 29, 15327
Otten, G. P. P. L., Vigan, A., Muslimov, E., et al. 2021, A&A, 646, A150
Paschall, R. N., & Anderson, D. J. 1993, Appl. Opt., 32, 6347
Paul, B., Sauvage, J.-F., & Mugnier, L. 2013, A&A, 552, A48
Pelt, D. M., & Sethian, J. A. 2018, Proc. Natl. Acad. Sci., 115, 254
Pou, B., Ferreira, F., Quinones, E., Gratadour, D., & Martin, M. 2022, Opt. Express, 30, 2991
Poyneer, L. A., & Macintosh, B. 2004, J. Opt. Soc. Am. A, 21, 810
Poyneer, L., & Véran, J.-P. 2008, J. Opt. Soc. Am. A, 25, 1486
Poyneer, L. A., Macintosh, B. A., & Véran, J.-P. 2007, J. Opt. Soc. Am. A, 24, 2645
Poyneer, L., van Dam, M., & Véran, J.-P. 2009, J. Opt. Soc. Am. A, 26, 833
Ragazzoni, R. 1996, J. Mod. Opt., 43, 289
Ronneberger, O., Fischer, P., & Brox, T. 2015, in International Conference on Medical Image Computing and Computer-assisted Intervention (Berlin: Springer), 234
Sinquin, B., Prengère, L., Kulcsár, C., et al. 2020, MNRAS, 498, 3228
Snellen, I., de Kok, R., Birkby, J. L., et al. 2015, A&A, 576, A59
Sun, Z., Chen, Y., Li, X., Qin, X., & Wang, H. 2017, Opt. Commun., 382, 519
Swanson, R., Lamb, M., Correia, C., Sivanandam, S., & Kutulakos, K. 2018, SPIE, 10703, 107031F
Swanson, R., Lamb, M., Correia, C. M., Sivanandam, S., & Kutulakos, K. 2021, MNRAS, 503, 2944
van Kooten, M., Doelman, N., & Kenworthy, M. 2017, Performance of AO predictive control in the presence of non-stationary turbulence (Instituto de Astrofisica de Canarias)
van Kooten, M., Doelman, N., & Kenworthy, M. 2019, J. Opt. Soc. Am. A, 36, 731
Vérinaud, C. 2004, Opt. Commun., 233, 27
Wong, A. P., Norris, B. R., Tuthill, P. G., et al. 2021, J. Astron. Teles. Instrum. Syst., 7, 019001
Xu, Z., Yang, P., Hu, K., Xu, B., & Li, H. 2019, Appl. Opt., 58, 1998

Publication III

Nousiainen, J., Engler, B., Kasper, M., Helin, T., Heritier, C. T., & Rajani, C.

Advances in model-based reinforcement learning for adaptive optics control

Reprinted with permission from
Adaptive Optics Systems VIII
Vol. 12185, pp. 882-891, 2022

Advances in model-based reinforcement learning for Adaptive Optics control

Jalo Nousiainen[a], Byron Engler[b], Markus Kasper[b], Tapio Helin[a], Cédric T. Heritier[b], and Chang Rajani[c]

[a]LUT University, Yliopistonkatu 34, FI-53850, Lappeenranta, Finland
[b]European Southern Observatory, Karl-Schwarzschild-Str. 2, 85748, Garching bei München, Germany
[c]University of Helsinki, Yliopistonkatu 4, FI-00100 Helsinki, Finland

ABSTRACT

Direct imaging of Earth-like exoplanets is one of the significant scientific drivers of the next generation of ground-based telescopes. Typically, Earth-like exoplanets are located at tiny angular separations from their host stars rendering their identification difficult. Consequently, the adaptive optics (AO) system's control algorithm must be carefully designed to distinguish the exoplanet from the residual light produced by the host star.

A new promising avenue of research aimed at improving AO control builds on data-driven control methods such as Reinforcement Learning (RL) methods. It is an active branch of the machine learning research field, where control of a system is learned through interaction with the environment. Thus, RL can be seen as an automated approach for AO control. In particular, model-based reinforcement learning (MBRL) has been shown to cope with both temporal and misregistration errors. Similarly, it has been demonstrated to adapt to non-linear wavefront sensing while being efficient to train and execute.

In this work, we implement and adapt an RL method called Policy Optimizations for AO (PO4AO) to the GHOST test bench at ESO headquarters, where we show strong performance on cascaded AO system lab simulation. Further, the results align with the previously obtained results with the method.

Keywords: adaptive optics, high contrast imaging, reinforcement learning

1. INTRODUCTION

High contrast imaging (HCI) is an imaging technique that combines eXtreme Adaptive Optics (XAO) with coronagraphy to produce images of faint sources located near bright point sources such as exoplanets next to their host stars. With current HCI instruments, direct images of exoplanets have been mostly limited to about a dozen very young and luminous giant exoplanets.[1–3] However, more planets could be directly imaged if the sensitivity close to the host star is improved, and the performance of the XAO system is the main limiting factor of this sensitivity close to the host star.

In HCI, for imaging close to the star, the main limiting factor of the performance of a well-tuned adaptive optics (AO) system is photon noise, and temporal error.[4] The temporal delay error of AO systems controlled by standard methods arises from wavefront sensor detector integration, detector readout, computation of the correction signal, and its application to the DM. This delay amounts to at least two AO system operating cycles. During this time window, atmospheric turbulence has evolved and no longer perfectly matches the DM correction.

The temporal delay error can be dampened in two ways: either by raising the operating frequency of the AO system or utilizing predictive control. The acceleration of the AO system can be achieved by installing the so-called second-stage downstream from a classical first-stage AO system.[5] The second-stage system only observes the residual from the first-stage AO system and, operating independently from the first-stage, can utilize DMs that can handle faster control loops. This approach is, for example, proposed for the upgrade of SPHERE (called SPHERE+;[6]) and is expected to deliver significantly improved raw point-spread function (PSF) contrast near the star.

Further author information: (Send correspondence to J.N.)
E-mail: jalo.nousiainen@lut.fi

Adaptive Optics Systems VIII, edited by Laura Schreiber, Dirk Schmidt, Elise Vernet,
Proc. of SPIE Vol. 12185, 121852O · © 2022 SPIE
0277-786X · doi: 10.1117/12.2630317

Lately, predictive control algorithms have gained significant attention in the field of HCI instrumentation. Remarkable progress has been achieved with various different approaches,[7–30] some methods have also been tested in laboratory setups or on-sky.[31] The advantage of predictive control is its ability to denoise non-correlated WFS measurements along with temporal error corrections. On the other hand, a classically controlled faster AO loop is always limited by the trade-off between photon noise and temporal error. However, if the algorithm is fast enough, it is possible to operate the faster second-stage system with predictive control.

This paper discusses a purely data-driven predictive control algorithm called the Policy Optimizations for AO (PO4AO) and the prospect of operating a second-stage AO system with it. A significant advantage of fully data-driven control, such as PO4AO, is that it does not require an estimate of the system's open-loop temporal evolution. Hence, it is insensitive to pseudo-open-loop reconstruction errors such as the optical gain effect, and misregistration.[29,32,33] Our contributions are three-fold: first, we operate PO4AO on the GPU-based High-order adaptive OpticS (GHOST) bench, which simulates a second-stage AO system by running numerically simulated residual turbulence phase screens across a programmable Spatial Light Modulator (SLM). Second, we introduce refinements to the original algorithm and, third, derive tuned hyper-parameters and analyze corresponding results against a well-tuned integrator controller. Further, we give a short discussion on future work.

2. CLASSICAL ADAPTIVE OPTICS CONTROL

An AO system is commonly controlled with a linear proportional integrator (PI) controller, later referred simply as the integrator. We consider it here as our reference method against the PO4AO as it is still widely used in AO. The integrator relies on a linear approximation of so-called interaction matrix mapping DM commands to WFS measurements, i.e., we have that

$$\Delta \boldsymbol{w}^t = Dv^t + \xi_t, \tag{1}$$

where $\Delta \boldsymbol{w}^t = (\delta w_1^t, \delta w_2^t, \cdots, \delta w_n^t)$ is the WFS data, v^t the DM commands and D is the interaction matrix and ξ_t is the measurement noise typically composed of photon and detector noise.

Once the interaction matrix is estimated, the inverse problem, i.e., reconstruction v^t given $\Delta \boldsymbol{w}^t$, needs to be considered. As D is generally not invertible, some regularization approach is needed. Here, we restrict ourselves to linear methods described by a reconstruction matrix C mapping WFS measurements to DM commands. As our regularization method we project D to a smaller dimensional subspace spanned by Karhunen–Loéve (KL) modal basis.[34] Each KL mode in the basis has a representation in terms of actuator voltages. This relation is fully determined by a transformation matrix B_m mapping DM actuator voltages to m first modal coefficients. The regularized reconstruction matrix is now defined by the Moore–Penrose pseudo-inverse

$$C_m = (DP_m)^\dagger, \tag{2}$$

where $P_m = B_m^\dagger B_m$ is a projection map to the KL basis. The role of m is to improve stability at the cost of resolution; smaller m results in lower noise amplification while producing a reconstruction with less modal basis functions. An optimal m balances the error produced by these two effects.

Let us now define the integrator for AO control. At a given time step t, the WFS measures the residual wavefront. The new control voltages $\tilde{\boldsymbol{v}}^t$ are obtained from

$$\tilde{\boldsymbol{v}}^t = \tilde{\boldsymbol{v}}^{t-1} + gC_m\Delta \boldsymbol{w}^t, \tag{3}$$

where g is the integrator gain, typically fixed below a value of about 0.5 for a two-step delay system.[35]

3. ADAPTIVE OPTICS AS A MARKOV DECISION PROCESS

In RL the AO control loop is modelled as a Markov decision process (MDP). As the WFS data $\boldsymbol{w}^t$ does not fully identify v^t, the AO control must be considered as a partially observed MDP.[26,27,32,33] However, the state space can be expanded to include a history WFS measurement and DM control voltages to guarantee approximately markovian statistics. This paper follows the expanded state space approach.

Let us denote the control voltages applied to DM at a given time instance t by $\tilde{\boldsymbol{v}}_t$ and the preprocessed WFS measurements by $\boldsymbol{w}_t$. We define the set of actions as the differential control voltages

$$\boldsymbol{a}_t = \Delta\tilde{\boldsymbol{v}}_t. \tag{4}$$

The state $\boldsymbol{s}_t$ of the ordinary (not partially observed) MDP is set according to

$$\boldsymbol{s}_t = \left(\boldsymbol{o}_t, \boldsymbol{o}_{t-1}, \ldots, \boldsymbol{o}_{t-k}, \boldsymbol{a}_{t-1}, \boldsymbol{a}_{t-2}, \ldots, \boldsymbol{a}_{t-k}\right), \tag{5}$$

where $\boldsymbol{o}_t = C_m \boldsymbol{w}_t$, i.e., the wavefront measurement projected to DM space and k the number of history frames used in state formulation.

For a state-action pair, the reward is the residual voltages' negative squared norm corresponding to the following measurement

$$r(\boldsymbol{s}_t, \boldsymbol{a}_t) = -\|\tilde{\boldsymbol{o}}_{t+1}\|^2, \tag{6}$$

where $\tilde{\boldsymbol{o}}_{t+1}$ is the next wavefront measurement projected to DM space. For more details of these choices, see Nousiainen et al.[33]

4. MODEL-BASED POLICY OPTIMIZATION

This section gives a brief description of PO4AO. The key idea is to learn a non-linear control law considered as the mapping from past telemetry to new DM commands from data collected from the AO loop. In RL, we note that this approximation is referred to as the *policy* and will be formulated as a mapping from the current state $\boldsymbol{s}_t$ to the next action $\boldsymbol{a}_t$. In AO terms, the policy is a controller combining the reconstruction and control law (e.g., a least-squares modal reconstruction followed by integrator control).

In this work the policy is constructed as a neural network and its parameters are specified indirectly via model-based policy optimizations. More precisely, the method learns so-called *dynamics model* that predicts the subsequent wavefront sensor measurement given the previous measurements and actions and uses this dynamics approximation to optimize the policy.[33] The method iterates following three phases:

1. **Running the policy:** the method collects data by running the policy in the AO control loop for T timesteps (a single episode).
2. **Improving the dynamics model:** the dynamics model parameter are optimized via a supervised learning objective.
3. **Improving the policy:** the policy parameters are optimized by utilizing the dynamics model.

Let us now describe this process in more detail. In PO4AO the dynamics model $\hat{p}_\omega : (\boldsymbol{s}_t, \boldsymbol{a}_t) \mapsto \boldsymbol{s}_{t+1}$ parametrized by ω is expressed as an ensemble, i.e., a collection of deterministic convolutional neural networks (CNNs), where ω represents the weights and biases of the networks. Moreover, the policy mapping $\pi_\theta : \boldsymbol{s}_t \mapsto \boldsymbol{s}_{t+1}$ parametrized by θ is constructed as a fully CNN followed by modal filter layer, i.e.,

$$\pi_\theta(\boldsymbol{s}_t) = P_m F_\theta(\boldsymbol{s}_t), \tag{7}$$

where P_m is the projection map to the KL basis defined above and F_θ is a fully CNN, where the output is vectorized. Again, the parameter θ represents the weights and biases of the CNN.

In **Step 1**, telemetry (tuples of $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ saved into a dataset $\mathcal{D}$) is collected by operating the AO control loop with the current (or initial) parametrization of the policy map.

Utilizing this data, in **Step 2**, the dynamics model is trained, i.e., the parametrization is optimized by minimizing the squared difference between the true next states and the predictions according to

$$\sum_{\mathcal{D}} \|\boldsymbol{s}_{t+1} - \hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)\|^2 = \sum_{\mathcal{D}} \|\boldsymbol{o}_{t+1} - \hat{\boldsymbol{o}}_{t+1}\|^2 \,, \tag{8}$$

where $\boldsymbol{o}_{t+1}$ is obtained from the state $\boldsymbol{s}_{t+1}$ and $\hat{\boldsymbol{o}}_{t+1}$ is the observation predicted by $\hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The parameters are optimized by the Adam algorithm.[36]

The objective of **Step 3** is to find policy parameters θ that maximize the expected reward within some pre-defined time horizon H given the dynamics of the environment (in our case the approximate model $\hat{p}_\omega$), that is

$$\arg\max_\theta \sum_{\mathrm{s} \in \mathcal{D}} \sum_{t=1}^{H} \hat{r}_\omega(\tilde{s}_t, \pi_\theta(\tilde{s}_t)), \tag{9}$$

where H is so-called *planning horizon* and

$$\tilde{\mathbf{s}}_1 = \mathbf{s} \quad \text{and} \quad \tilde{\mathbf{s}}_{t+1} = \hat{p}_\omega(\tilde{s}_t, \pi_\theta(\tilde{s}_t)).$$

In practice, this is carried out by sampling from previously observed data points, computing the actions and using the dynamics model to simulate the future. Moreover, we use the differentiability of the reward, and backpropagate through the models. Algorithm 1 gives a full pseudo-code for the procedure. For more details of the method, see Nousianen et al.[33]

Algorithm 1 Policy Optimization for Adaptive Optics (PO4AO)

```
Initialize policy and dynamics model parameters $\theta$ and $\omega$ randomly
Initialize gradient iteration length $K$, batch size $B < |\mathcal{D}|$ and planning horizon $H$
while not converged do
    Generate samples $\{s_{t+1}, s_t, a_t\}$ by running policy $\pi_\theta(a_t|s_t)$ for $T$ timesteps (an episode) and append to $\mathcal{D}$
    Fit dynamics by minimizing Eq. (8) w.r.t $\omega$ using Adam
    for iteration $k = 1$ to $K$ do
        Sample a mini batch of $B < |\mathcal{D}|$ states $\{s_\tau\}$ from $\mathcal{D}$
        for each $\mathbf{s}_\tau$ in the mini batch do
            Set $\tilde{\mathbf{s}}_1^\tau = \mathbf{s}_\tau$
            for $t = 1$ to $H$ do
                Predict $\boldsymbol{a}_t = \pi_\theta(\boldsymbol{s}_t)$
                Predict $\boldsymbol{s}_{t+1} = \hat{p}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$
                Calculate $R_t = \hat{r}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t)$
            end for
        end for
        Update $\theta$ by taking a gradient step according to $\nabla_\theta \sum_{t=\tau}^{\tau+H} R_t$ with Adam.
    end for
end while
```

4.1 PO4AO for GHOST

Let us describe some modifications applied here to the algorithm presented in Nousiainen et al.[33] We found that the original algorithm occasionally produced oscillations on the DM, especially in the early stage of the training. As a remedy, we introduced small regularizing term to the reward function used in policy optimization. More precisely, we added a small reward term to favor small actions such that

$$\hat{r}_\omega(\boldsymbol{s}_t, \boldsymbol{a}_t) = -||\tilde{\boldsymbol{o}}_{t+1}||^2 - \alpha||\boldsymbol{a}_t||^2. \tag{10}$$

This modification stabilized the training procedure and no oscillations were observed on the DM afterwards. The value $\alpha = 0.1$ provided enough regularization without affecting the performance.

Another observation was related to DM saturation. The POAO saturated the DM while learning the system by increasing the norm of high-order modes outside the KL basis. Since the policy output is filtered with the KL basis, it cannot recover these high-order modes. We solved this problem by implementing a standard *garbage collector* in the loop,[37] that is, we clean the modes outside our control radius from the full control voltages applied to the mirror.

5. RESULTS

5.1 Simulations set-up

We implemented PO4AO to the GHOST instrument, located in ESO headquarters in Garching, Germany, and compared it against a well-tuned integrator. GHOST is an experimental adaptive optics system to explore new AO control methods (particularly predictive control) for the ELT Planetary Camera and Spectrograph (PCS). It is a simple single-source on-axis system equipped with a pyramid WFS and DM Boston Micromachines (BMC) DM-492 deformable mirror. A programmable Spatial Light Modulator (SLM) injects turbulence with HD resolution. We experimented with two temporal delays of two and three frames. The algorithm remains the same for both experiments - it automatically adapts to temporal delay.

The GHOST was used to simulate a faster second-stage system as follows. First, we numerically generated the residual phase screens for the lab setup. We simulated an 8-meter telescope with 41x41 DM and a PWFS observing a 6.16 magnitude natural guide star. Atmospheric turbulence was simulated as a sum of nine frozen flow layers with Von Karman power spectra combining a Fried parameter r_0 of 15 cm at 550 nm wavelength. The simulation parameters are listed in Table 1. We controlled the simulated system with an integrator and recorded the residual turbulence after DM correction. Then, we replayed these residual phase screens with the SLM during the GHOST experiments.

We set the episode length (the rate of model updates) to 500 frames and the number of history frames to 20. The number of history frames was tuned to give to maximize rewards. Both experiments started with measuring the interaction matrix by poking the DM actuators (inside the linear range of the WFS) and recording the corresponding WFS measurements. More precisely, we calibrate the system with Hadamard patterns,[38] and the reconstructions matrix and the KL-filter were calculated with 300 KL modes.

We let both experiments run for 50 episodes (i.e., 25 000 frames in total); see Figs. 1a and 1b. After each episode, the simulation is halted for training the dynamics and policy. The first ten episodes execute so-called *warm-up* for the policy (collected initial data-set and trained the models) with noisy integrator control, i.e., integrator with some added noise in the control voltages. The noise is reduced linearly after each episode such that the first episode was run with high binary noise and the 10th episode with zero noise (see the dashed gray line in Figs. 1a and 1b).

With the given hyperparameters, hardware, and implementation, training (after each episode) took 0.72 seconds. In practice, the training procedure should be implemented parallel to control, enabling a ”real-time” version of the method. After the 50 episodes, we ran the policy for 6 000 extra frames to record the science camera PSF and compare it to PSF obtained with the integrator over a similar time interval; see Fig 2. We tuned the integrator gain manually for both time delays.

5.2 PO4AO performance

We evaluate the performance of PO4AO in rewards (proportional to the WFS slopes variance) and with the science camera images. Figures 1a and 1b show both the training speed and the performance (in rewards) of the method. The PO4AO outperforms the integrator after the warm-up, i.e., the first 5000 frames, and improves a bit until 40 episodes. It provides a factor of 2.4 improvement in reward compared to the integrator for the 2-frame delay experiment and a factor of 2.6 for 3-frame delay experiment.

We further evaluate PO4AO's performance with the science camera image. The coronagraph was not installed during the experiments; thus, the airy rings dominate the images. However, we can still see the two control radius' (outer from the numerical simulation and inner from the GHOST second-stage control) nicely as well as some temporal halo, i.e., e., the extra light in the direction of the dominant wind (horizontal). For both time delays, we see that PO4AO lowers the light intensity inside the inner control radius, especially in the horizontal direction, making the inner control radius more clearly visible. We expect the improvement to be more visible once the coronagraph is installed and well-aligned.

The real-time frame rate of the experiment was 100Hz. We had to trigger the SLM update through a remote server, and that set the limiting factor for the loop speed. With the current python-based PyTorch implementation, the policy model added 0.68 ms of extra inference time at every timestep. As discussed before, the training of the models could be implemented in parallel to control and, hence, would not theoretically add any extra time to the procedure.

6. DISCUSSION AND FUTURE WORK

To conclude, we implemented PO4AO to the GHOST test bench and introduced the slight modifications needed for the setup. We showed that the method is fully applicable for cascaded AO systems, where the algorithm controls the second stage. The results show strong performance in terms of rewards and training speed. We also compared the science camera images (dominated by the airy rings) and observed an improvement. However, we expect more significant improvement once the coronagraph is installed and well-aligned. Further, all the results also align well with previous results on the methods.[33]

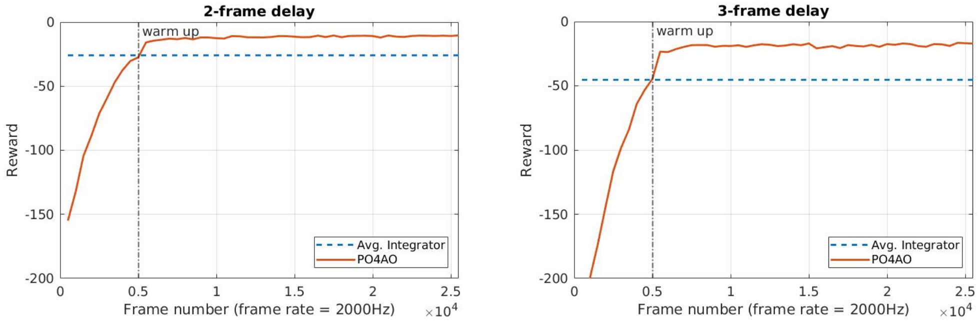
Table 1: Simulations parameters

Numerically simulated first-stage		
Parameter	Value	Units
Telescope diameter	8	m
Obstruction ratio	0	percent
Sampling frequency	1000	Hz
NGS magnitude	6.16	···
WFS wavelength	0.79	µm
Actuators	41	across the pupil
PWFS modulation	3	λ/D
KL modes	900	modes
Integrator gain	0.5	···
GHOST (second-stage)		
Parameter	Value	Units
Sampling frequency (simulation)	2000	Hz
Sampling frequency (real-time)	100	Hz
Actuators	24	across the pupil
PWFS modulation	4	λ/D
KL modes	300	modes
Atmosphere parameters		
Fried parameter	15	cm @ 500 nm
Number of layers	9	···
C_N^2	0.52 / 0.19 / 0.07 / 0.06 / 0.03 / 0.04 / 0.04 / 0.03 / 0.02	percent (%)
Average wind speeds	34	m/s
L_0 (m)	30	m
PO4AO parameters		
Planning horizon (H)	4	steps
Past DM commands (m)	20	commands
Past WFS measurements (k)	20	frames
CNN ensemble size	5	···
Dynamics iterations / episode	30	steps
Policy iterations / episode	18	steps
Training mini batch size	32	···
CNN parameters		
Number of conv. layers	3	layers
Number of filt./layers	64	filters
Filter size	3 × 3	pixels
Padding	1	pixels
Activation functions	Leaky ReLU	···

Further development and research are needed, most notably, in two directions. First, we need to implement (preferably with a lower lever programming language) the training procedure parallel to control to enable the ”real-time” training. Second, the atmospheric statistics were fixed during the experiments. Future work should address maintaining the best possible performance under varying turbulence conditions.

ACKNOWLEDGMENTS

JN was supported by the ESO. The work of TH was supported by the the Academy of Finland (decisions 320082 and 326961).



(a)

(b)

Figure 1: Training plots for the GHOST experiments. *Panel a* is for 2 frame delay in the system and *Panel b* for 3 frame delay. The red lines correspond to performance of PO4AO during each episode and blue dashed lines for average integrator performance. The gray dashed line marks the end of integrator warm up for PO4AO. In both experiments, the PO4AO outperforms the integrator in reward (proportional to WFS slopes variance) all ready after the warm up period.

REFERENCES

[1] Marois, C., Zuckerman, B., Konopacky, Q. M., Macintosh, B., and Barman, T., "Images of a fourth planet orbiting hr 8799," *Nature* **468**(7327), 1080–1083 (2010).

[2] Lagrange, A.-M., Gratadour, D., Chauvin, G., Fusco, T., Ehrenreich, D., Mouillet, D., Rousset, G., Rouan, D., Allard, F., Gendron, É., et al., "A probable giant planet imaged in the β pictoris disk-vlt/naco deep l'-band imaging," *Astronomy & Astrophysics* **493**(2), L21–L25 (2009).

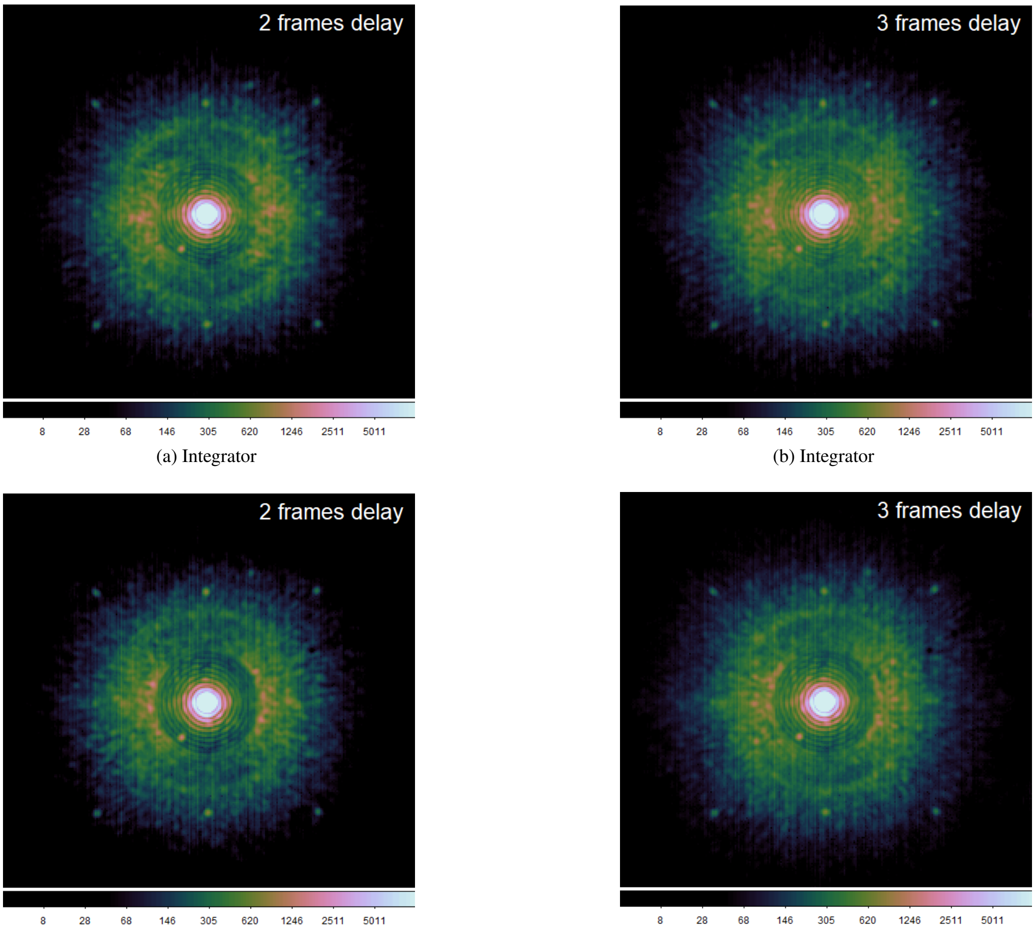
[3] Macintosh, B., Graham, J., Barman, T., De Rosa, R., Konopacky, Q., Marley, M., Marois, C., Nielsen, E., Pueyo, L., Rajan, A., et al., "Discovery and spectroscopy of the young jovian planet 51 eri b with the gemini planet imager," *Science* **350**(6256), 64–67 (2015).

[4] Guyon, O., "Limits of adaptive optics for high-contrast imaging," *The Astrophysical Journal* **629**(1), 592 (2005).

[5] Cerpa-Urra, N., Kasper, M., Kulcsár, C., Raynaud, H.-F., and Heritier, C. T., "Cascade adaptive optics: contrast performance analysis of a two-stage controller by numerical simulations," *Journal of Astronomical Telescopes, Instruments, and Systems* **8**(1), 019001 (2022).

[6] Boccaletti, A., Chauvin, G., Mouillet, D., Absil, O., Allard, F., Antoniucci, S., Augereau, J.-C., Barge, P., Baruffolo, A., Baudino, J.-L., Baudoz, P., Beaulieu, M., Benisty, M., Beuzit, J.-L., Bianco, A., Biller, B., Bonavita, B., Bonnefoy, M., Bos, S., Bouret, J.-C., Brandner, W., Buchschache, N., Carry, B., Cantalloube, F., Cascone, E., Carlotti, A., Charnay, B., Chiavassa, A., Choquet, E., Clenet, Y., Crida, A., De Boer, J., De Caprio, V., Desidera, S., Desert, J.-M., Delisle, J.-B., Delorme, P., Dohlen, K., Doelman, D., Dominik, C., Orazi, V., Dougados, C., Doute, S., Fedele, D., Feldt, M., Ferreira, F., Fontanive, C., Fusco, T., Galicher, R., Garufi, A., Gendron, E., Ghedina, A., Ginski, C., Gonzalez, J.-F., Gratadour, D., Gratton, R., Guillot, T., Haffert, S., Hagelberg, J., Henning, T., Huby, E., Janson, M., Kamp, I., Keller, C., Kenworthy, M., Kervella, P., Kral, Q., Kuhn, J., Lagadec, E., Laibe, G., Langlois, M., Lagrange, A.-M., Launhardt, R., Leboulleux, L., Le Coroller, H., Li Causi, G., Loupias, M., Maire, A., Marleau, G., Martinache, F., Martinez, P., Mary, D., Mattioli, M., Mazoyer, J., Meheut, H., Menard, F., Mesa, D., Meunier, N., Miguel, Y., Milli, J., Min, M., Molliere, P., Mordasini, C., Moretto, G., Mugnier, L., Muro Arena, G., Nardetto, N., Diaye, M. N., Nesvadba, N., Pedichini, F., Pinilla, P., Por, E., Potier, A., Quanz, S., Rameau, J., Roelfsema, R., Rouan, D., Rigliaco, E., Salasnich, B., Samland, M., Sauvage, J.-F., Schmid, H.-M., Segransan, D., Snellen, I., Snik, F., Soulez, F., Stadler, E., Stam, D., Tallon, M., Thebault, P., Thiebaut, E., Tschudi, C., Udry, S., van Holstein, R., Vernazza, P., Vidal, F., Vigan, A., Waters, R., Wildi, F., Willson, M., Zanutta, A., Zavagno, A., and Zurlo, A., "Sphere+: Imaging young jupiters down to the snowline," *arXiv preprint arXiv:2003.05714* (2020).

[7] Kulcsár, C., Raynaud, H.-F., Petit, C., Conan, J.-M., and Lesegno, P. V. D., "Optimal control, observers and integrators in adaptive optics," *Optics express, 14(17):7464–7476* (2006).



(a) Integrator

(b) Integrator

(c) PO4AO

(d) PO4AO

Figure 2: Science camera images captured during the last 6000 frames of the experiments.*Panel a:* The integrator with 2-frame delay. *Panel b:* integrator with 3-frame delay *Panel c:* PO4AO with 2-frame delay *Panel d:*PO4AO with 2-frame delay.

[8] Paschall, R. N. and Anderson, D. J., "Linear quadratic gaussian control of a deformable mirror adaptive optics system with time-delayed measurements," *Applied optics* **32**(31), 6347–6358 (1993).

[9] Gray, M. and Le Roux, B., "Ensemble transform kalman filter, a nonstationary control law for complex ao systems on elts: theoretical aspects and first simulations results," in [*Adaptive Optics Systems III*], **8447**, 84471T, International Society for Optics and Photonics (2012).

[10] Conan, J.-M., Raynaud, H., AR, Kulcsár, C., Meimon, S., and Sivo, G., "Are integral controllers adapted to the new era of elt adaptive optics?," in [*AO4ELT*], (2011).

[11] Correia, C., Conan, J.-M., Kulcsár, C., Raynaud, H.-F., and Petit, C., "Adapting optimal lqg methods to elt-sized ao systems," in [*1st AO4ELT conference-Adaptive Optics for Extremely Large Telescopes*], 07003, EDP Sciences (2010).

[12] Correia, C., Raynaud, H.-F., Kulcsár, C., and Conan, J.-M., "On the optimal reconstruction and control of adaptive optical systems with mirror dynamics," *JOSA A* **27**(2), 333–349 (2010).

[13] Correia, C. M., Bond, C. Z., Sauvage, J.-F., Fusco, T., Conan, R., and Wizinowich, P. L., "Modeling astronomical adaptive optics performance with temporally filtered wiener reconstruction of slope data," *JOSA A* **34**(10), 1877–1887 (2017).

[14] Sinquin, B., Prengère, L., Kulcsár, C., Raynaud, H.-F., Gendron, E., Osborn, J., Basden, A., Conan, J.-M., Bharmal, N., Bardou, L., et al., "On-sky results for adaptive optics control with data-driven models on low-order modes," *Monthly Notices of the Royal Astronomical Society* **498**(3), 3228–3240 (2020).

[15] Guyon, O. and Males, J., "Adaptive optics predictive control with empirical orthogonal functions (eofs)," *arXiv preprint arXiv:1707.00570* (2017).

[16] Poyneer, L. A., Macintosh, B. A., and Véran, J.-P., "Fourier transform wavefront control with adaptive prediction of the atmosphere," *JOSA A* **24**(9), 2645–2660 (2007).

[17] Dessenne, C., Madec, P.-Y., and Rousset, G., "Optimization of a predictive controller for closed-loop adaptive optics," *Applied optics* **37**(21), 4623–4633 (1998).

[18] van Kooten, M., Doelman, N., and Kenworthy, M., [*Performance of AO predictive control in the presence of non-stationary turbulence*], Instituto de Astrofisica de Canarias (2017).

[19] van Kooten, M., Doelman, N., and Kenworthy, M., "Impact of time-variant turbulence behavior on prediction for adaptive optics systems," *JOSA A* **36**(5), 731–740 (2019).

[20] Swanson, R., Lamb, M., Correia, C., Sivanandam, S., and Kutulakos, K., "Wavefront reconstruction and prediction with convolutional neural networks," in [*Adaptive Optics Systems VI*], **10703**, 107031F, International Society for Optics and Photonics (2018).

[21] Sun, Z., Chen, Y., Li, X., Qin, X., and Wang, H., "A bayesian regularized artificial neural network for adaptive optics forecasting," *Optics Communications* **382**, 519–527 (2017).

[22] Liu, X., Morris, T., and Saunter, C., "Using long short-term memory for wavefront prediction in adaptive optics," in [*International Conference on Artificial Neural Networks*], 537–542, Springer (2019).

[23] Wong, A. P., Norris, B. R., Tuthill, P. G., Scalzo, R., Lozi, J., Vievard, S., and Guyon, O., "Predictive control for adaptive optics using neural networks," *Journal of Astronomical Telescopes, Instruments, and Systems* **7**(1), 019001 (2021).

[24] Males, J. R. and Guyon, O., "Ground-based adaptive optics coronagraphic performance under closed-loop predictive control," *Journal of Astronomical Telescopes, Instruments, and Systems* **4**(1), 019001 (2018).

[25] Swanson, R., Lamb, M., Correia, C. M., Sivanandam, S., and Kutulakos, K., "Closed loop predictive control of adaptive optics systems with convolutional neural networks," *Monthly Notices of the Royal Astronomical Society* **503**(2), 2944–2954 (2021).

[26] Pou, B., Ferreira, F., Quinones, E., Gratadour, D., and Martin, M., "Adaptive optics control with multi-agent model-free reinforcement learning," *Opt. Express* **30**, 2991–3015 (Jan 2022).

[27] Landman, R., Haffert, S. Y., Radhakrishnan, V. M., and Keller, C. U., "Self-optimizing adaptive optics control with reinforcement learning," in [*Adaptive Optics Systems VII*], **11448**, 1144849, International Society for Optics and Photonics (2020).

[28] Landman, R., Haffert, S. Y., Radhakrishnan, V. M., and Keller, C. U., "Self-optimizing adaptive optics control with reinforcement learning for high-contrast imaging," *Journal of Astronomical Telescopes, Instruments, and Systems* **7**(3), 039002 (2021).

[29] Haffert, S. Y., Males, J., Close, L., van Gorkom, K., Long, J., Hedglen, A., Schatz, L., Lumbres, J., Rodack, A., Knight, J., et al., "Data-driven subspace predictive control: lab and on-sky demonstration.," in [*Techniques and Instrumentation for Detection of Exoplanets X*], **11823**, 118231C, International Society for Optics and Photonics (2021).

[30] Haffert, S. Y., Males, J. R., Close, L. M., Van Gorkom, K., Long, J. D., Hedglen, A. D., Guyon, O., Schatz, L., Kautz, M. Y., Lumbres, J., et al., "Data-driven subspace predictive control of adaptive optics for high-contrast imaging," *Journal of Astronomical Telescopes, Instruments, and Systems* **7**(2), 029001 (2021).

[31] van Kooten, M. A., Jensen-Clem, R., Cetre, S., Ragland, S., Bond, C. Z., Fowler, J., and Wizinowich, P., "Predictive wavefront control on keck ii adaptive optics bench: on-sky coronagraphic results," *Journal of Astronomical Telescopes, Instruments, and Systems* **8**(2), 029006 (2022).

[32] Nousiainen, J., Rajani, C., Kasper, M., and Helin, T., "Adaptive optics control using model-based reinforcement learning," *Optics Express* **29**(10), 15327–15344 (2021).

[33] Nousiainen, J., Rajani, C., Kasper, M., Helin, T., Haffert, S., Vérinaud, C., Males, J., Van Gorkom, K., Close, L., Long, J., et al., "Towards on-sky adaptive optics control using reinforcement learning," *arXiv preprint arXiv:2205.07554* (2022).

[34] Gendron, E., "Modal Control Optimization in an Adaptive Optics System," in [*European Southern Observatory Conference and Workshop Proceedings*], *European Southern Observatory Conference and Workshop Proceedings* **48**, 187 (Jan. 1994).

[35] Madec, P.-Y., "Control techniques," *Adaptive optics in astronomy* , 131–154 (1999).

[36] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980* (2014).

[37] Petit, C., Meimon, S., Fusco, T., Kulcsár, C., and Raynaud, H.-F., "Hybrid lqg/integrator control for the vlt extrem ao system sphere," in [*2010 IEEE International Conference on Control Applications*], 878–883, IEEE (2010).

[38] Kasper, M., Fedrigo, E., Looze, D. P., Bonnet, H., Ivanescu, L., and Oberti, S., "Fast calibration of high-order adaptive optics systems," *Journal of the Optical Society of America A* **21**, 1004–1008 (June 2004).

Publication IV

Krokberg, T., Nousiainen, J., Lehtonen, J., & Helin, T.
FitAO: a Python-based platform for algorithmic development AO

Reprinted with permission from
Adaptive Optics Systems VIII
Vol. 12185, pp. 1031-1037, 2022

FitAO: a Python-based platform for algorithmic development in AO

Tomi Krokberg[a], Jalo Nousiainen[a], Jonatan Lehtonen[b], and Tapio Helin[a]

[a]LUT University, Yliopistonkatu 34, FI-53850, Lappeenranta, Finland
[b]University of Helsinki, Yliopistonkatu 4, FI-00100 Helsinki, Finland

ABSTRACT

We present FitAO, which is an open-source Python-based concept platform for algorithmic development in adaptive optics (AO). Control and reconstruction algorithms designed on FitAO can be executed simultaneously on multiple supported end-to-end simulation environments. It utilizes interface specifications of OpenAI Gym library enabling direct access to an extensive set of control algorithms. With these properties, FitAO aims to facilitate comparative studies of AO control and reconstruction algorithms, and pave the way for modern data-driven and hybrid algorithms. We provide a brief tutorial example and discuss future development.

Keywords: adaptive optics, simulation, reinforcement learning

1. INTRODUCTION

Development of control and reconstruction algorithms plays an important role for the success of adaptive optics (AO) systems in the ELT generation telescopes due to the vast increase in data flow and complexity of system designs. Supporting this effort, a number of end-to-end simulation environments implementing full AO systems in software have emerged during the last 15 years. For the research community, such simulation environments typically provide open-source access to basic AO systems and functionality with varying degrees of support from the research group or consortium behind the development. Moreover, the programming language used for implementation varies based on the background and focus of the developer. As examples of such simulation environments let us mention the C/Python-based Compass[1] and DASP,[2] Python-based SOAPY,[3] Matlab-based OOMAO[4] and MOST,[5] Yorick-based YAO[6] and IDL-based CAOS.[7]

Such a rich library of simulation software provides flexibility for AO researchers while the platform divergence has given rise to some unwanted side-effects as well. The major disadvantage as experienced by the authors is that while the number of different algorithmic solutions has rapidly grown, comparative studies have not emerged as steadily making it challenging to identify the state-of-the-art. This is understandable as algorithmic implementations in AO are traditionally not made openly available and even when they are, the further effort to port the implementation to new environment may not be straightforward. In particular, open benchmarking such as in the machine learning community is not a common standard in AO and would be a very welcome development in future.

Another aspect clouding efforts of comparison is that while simulation environments aim for an ideal execution of the AO system, some components of the physics or system can be modelled differently leading to possibly minor but inherent differences in output. In the same vein, some system or model parameters (consider the magnitude of the guide star as a trivial example) are simply interpreted differently requiring careful comparison of the results. Be it as it may, comparative studies are destined to become more complicated in the near future with the emerging wave of data-driven algorithms in AO. Namely, the quality and amount of the data together with the training process will affect the performance and need to be taken into account.

With our FitAO software package we propose a step towards more straightforward comparative studies in the AO research landscape dominated (in future) by data-driven or hybrid algorithms. The main idea of FitAO is to render control and reconstruction algorithmic development independent of the simulation environments by providing a Python-based platform with two fundamental properties: first, any algorithm developed on FitAO can be executed on various end-to-end simulation environments (at this stage, interface to OOMAO, Compass and Soapy is provided) due to the tools in Python provided to wrap software such as MATLAB. Second, FitAO utilizes the interface specifications of OpenAI Gym,[8] which

Email: jalo.nousiainen@lut.fi

Adaptive Optics Systems VIII, edited by Laura Schreiber, Dirk Schmidt, Elise Vernet,
Proc. of SPIE Vol. 12185, 1218539 · © 2022 SPIE
0277-786X · doi: 10.1117/12.2630300

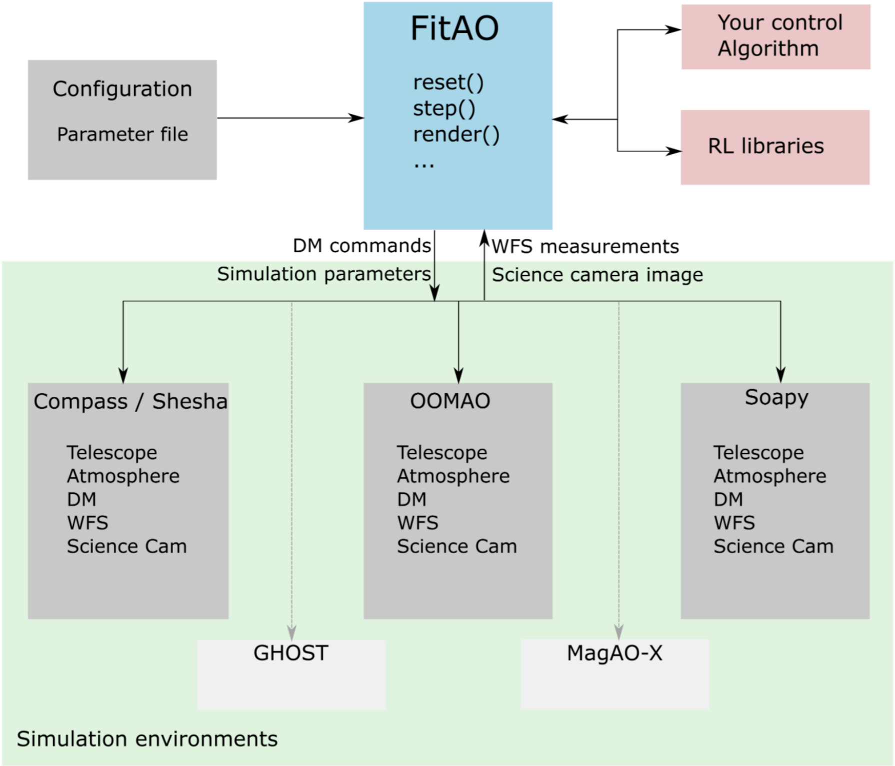


Figure 1: Conceptual visualization of FitAO.

enables direct integration to the various RL algorithm libraries designed with the same specifications. FitAO is open-source and is available at `http://github.com/jnousi/FitAO/`.

Our aim with the first property is to reduce dependency on the simulation software when evaluating performance, to enable easier comparison e.g. with results appearing in literature and, perhaps most importantly, to render collaboration more straightforward by reducing the effort needed in transitioning to another simulation software, optical bench or instrument. The second property integrates recent developments in particular related to reinforcement learning and control more intimately to AO, and paves the way for easier implementation of data-driven methods.

An early success story illustrating these properties was carried out by the authors, who developed in Nousiainen et al.[9] a novel reinforcement learning algorithm for the control task in extreme AO systems. Using early versions of FitAO this algorithm was improved and tested on the MagAO-X instrument.[10] More recently, the algorithm was implemented on the GHOST test bench operated by the ESO utilizing a similar interface.[11]

This article is organized as follows: in section 2 we discuss functionality and main components of FitAO. We demonstrate the use and capabilities in section 3. Future development is discussed in section 4 and we give conclusions in section 5.

2. FUNCTIONALITY

In this section we discuss functionality of the FitAO package. Conceptual functionality is summarized in Figure 1.

2.1 Interface to simulation environments

The key idea in FitAO is that each simulation tool is introduced as an *RL environment* according to the OpenAI Gym specifications. Each Gym environment needs a predefined set of basic functions that enable integration to an OpenAI Gym library such as Stable-baselines. Gym environment enables the use of wrappers,[12] which make it possible to change the observations seen and the actions made by the RL algorithm, e.g. change the control space to a subset of modal commands or preprocess the WFS measurement. These wrappers execute suitable commands in the simulation environment.

For minimum viable functionality FitAO has to be able to access procedures for setting the system parameters, returning measured slopes and setting deformable mirror commands. These functions can often be accessed using Python, even if the original language differs from it. For example, in the current version of FitAO the Matlab-based OOMAO environment is operated through the Matlab Engine API provided by Matlab for Python. Python bindings also exist for C/C++, CUDA and many other common languages, enabling operation of most simulation environments.

Let us now briefly describe most relevant functions specified in the Gym interface and how they are interpreted in the AO context in FitAO.

```
step(action, showAtmos = True)
```

The `step`-function is used to evolve the simulation forward and interact with the telescope. `action` is the command given to the DM, which can be delayed by a buffer inside this function and scaled to enable robust generation of the interaction matrix. `showAtmos` controls whether the atmosphere is taken into account when observing the incoming wavefronts.

First, the step function sets the DM shape based on the delayed command. Then the wavefront is propagated through the atmosphere (if enabled) and DM. Note that the observed slopes from this propagation step are always at least one time-step behind the actual slopes to simulate the delays present in the system and scaled to match the commands. The step function returns the scaled slopes, a reward, a boolean whether the RL episode is finished and a dictionary of other important info.

```
reset(seed = -1)
```

This function is used to set a seed for the atmosphere generation enabling the use of the same atmosphere multiple times. `Reset` sets DM commands to zero, initializes atmosphere with current seed and returns the scaled slopes after propagation.

```
render(mode='rgb_array')
```

This routine is used to visualize the current state in the system. Currently it plots WFS residual wavefront, the DM shape, combined atmosphere, target image, and raw WFS image. `mode` is currently unused, but it could be used to set some predefined modes of rendering.

```
set_params_file(param_file)
```

This routine communicates the simulation parameters to the simulation environment. The variable `param_file` gives the path of the parameter file to be used. This function also calls `set_params()` to set the parameters. `set_params_file` returns a boolean indicating whether the given parameter file defines a valid configuration for the simulation environment being used.

```
set_params(seed=None)
```

The atmosphere is generated based on the `seed` variable. At first, this routine sets the atmospheric parameters based on the parameter file. Next, the routine updates all relevant parameters and then calls for initialization of the simulation and defining of action and observation spaces according to the sizes of DM commands and slope measurements, respectively.

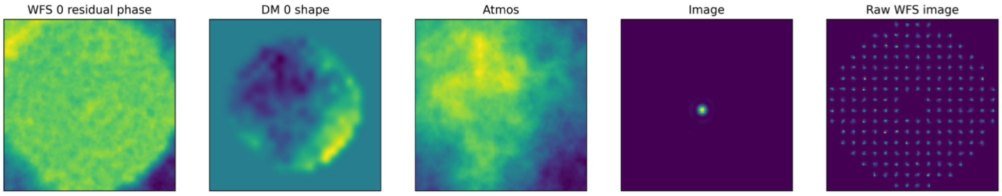


Figure 2: Plot produced by `render`-function to visualize the system state.

2.2 Configuration of simulation parameters

FitAO uses parameter files to define the simulation. Parameters which you can set currently include

- operation frequency and delay,
- telescope size, obstruction ratio and pupil diameter,
- common atmospheric parameters (such as number, altitudes and L_0 and C_n^2 profiles of the turbulence layers),
- DM specifications and
- Shack–Hartmann or Pyramid WFS subapertures, pixel and noise specification.

More environment-specific parameters can be added easily.

For each simulation environment, an internal routine is developed which translates and scales the FitAO parameter values properly for the environment. To ease the implementation of new simulation environments, FitAO currently provides tools to parse the parameter files into an easily accessible dictionary format, generates the common calibration matrices and includes basic implementations of integrator/pseudo-open loop controllers. As a result, implementing a new environment should be straightforward given that the fundamental commands in the environment are accessible by Python.

3. DEMONSTRATION

In this section we describe a tutorial example of using FitAO together with RL algorithms available in the Gym library Stable-baselines. The example code is available at http://www.github.com/jnousi/FitAO/.

3.1 Classical linear integrator control

Let us illustrate how to set up a simple AO simulation with the Compass simulation environment. The first step is to choose and import the desired environment. Each resides in its own folder and can be imported using a relative import. For example, the Compass environment can be imported with

```
from CompassEnv.CompassEnv import CompassEnv
```

This gives us access to the CompassEnv object, which allows interaction with the simulator and follows the OpenAI Gym interface implementing all the necessary functions.[13] It is possible to import multiple environments simultaneously, but note that operating multiple environments requires large computational resources. To setup the simulation, we must define the environment object in Python and give it a parameter file as follows

```
param_file = "./Conf/sh_16x8.py"
env = CompassEnv()
env.set_params_file(param_file)
```

Next we reset, initialize and start the simulation by using

```
    observation = env.reset()
```

This initiates the simulation and produces the first observation from the system, but can also be used to reset the simulation and start a new one. By default, the returned output is the measured slope data. The telescope is then operated with the step-function:

```
    observation, reward, done, info = env.step(action)
```

This function sets the new commands (action) and returns the next WFS measurement (observation), taking into account the desired control delay. The function also returns a reward for the action, which can be used for reinforcement learning, and a boolean value done to indicate to the RL algorithm whether the episode has ended or not. Furthermore, it also returns an info dictionary which can be used to return additional data such as the science camera image.

After setting these functions, we can calibrate the system and construct a standard integrator controller.

```
    import Tools.mat as tm
    import numpy as np

    S2V = tm.do_cmat(env,0.05)
    obs = env.reset()
    last_action = 0 * np.matmul(S2V,obs)

    gain = 0.5

    for i in range(n):
        action = last_action - gain * np.matmul(S2V,obs)
        last_action = action

        obs, reward, done, info = env.step(action, showAtmos=True)
```

3.2 Proximal policy optimization control

Let us now illustrate how to combine FitAO with the RL algorithm library through a simple control problem example. We study a closed-loop SCAO setup, where integrator control is applied. We utilize a popular reinforcement learning algorithm called proximal policy optimization (PPO)[14] from the Stable-baselines library [*] to adaptively tune the scalar integrator gain.

We start the demonstration by defining a wrapper that transitions from the full control problem to a reduced one, i.e, instead of controlling all actuators and observing full WFS images we are interested only in tuning the scalar gain. In what follows, the class AoGain will act as this wrapper, which will be available at the code repository. For more information about gym wrappers, see the documentation.[12]

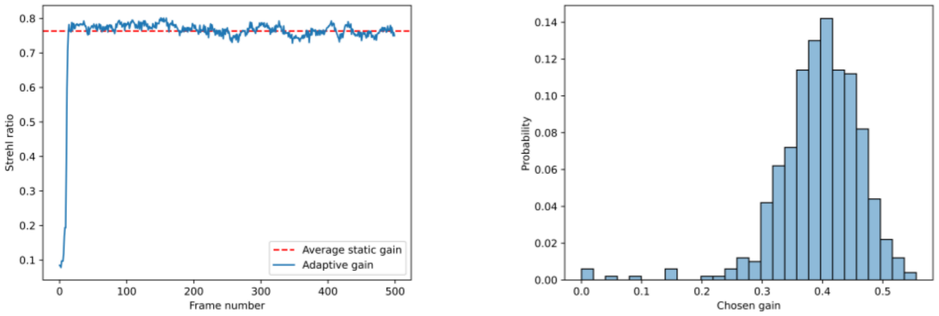
The following code initializes the simulation, wraps it using AoGain and applies the PPO algorithm. After training the model is saved into the file ppo2_AO.

```
from Tools.rl_gain import AoGain
from CompassEnv.CompassEnv import CompassEnv

from stable_baselines.common.policies import MlpPolicy
from stable_baselines import PPO2

param_file = "./Conf/sh_16x8.py"
env = AoGain(CompassEnv(),param_file)
```

[*] https://stable-baselines.readthedocs.io/en/master/



(a) Comparison between Strehl-performance of PPO with the average of static optimized gain.

(b) Histogram of the gain values chosen by the PPO algorithm.

Figure 3: Results on the PPO experiment.

```
model = PPO2(MlpPolicy, env, verbose=1, tensorboard_log="./tensorboard_gain/")
model.learn(total_timesteps=int(3e5))
model.save("ppo2_AO")
```

The learned model can be used to indicate the best action at each step with the following command

```
action, _states = model.predict(obs)
```

3.3 Results

We simulated an 8-meter telescope at 500 Hz equipped with a Shack–Harmann WFS and 17 x 17 DM observing a single natural guide star. The atmosphere was modelled with four layers below 9000 meters and with varying wind directions and speeds between 10 and 35 m/s. The outer scale was set to 25m and Fried parameter r_0 to 15cm. The detailed parameters are available in the FitAO repository.

An optimized static gain was compared to the PPO algorithm discussed in section 3.2. To stabilize the control with PPO we restrict the gain value to the interval $[-0.5, 0.5]$. Moreover, in the reward function we mildly penalize for large variations in gain.

In figure 2 the `render`-function has been used to show a visual presentation of the simulation at a given step. The performance of the PPO control is compared to an optimized static gain in figure 3. The same figure illustrates the variations of the gain plotted in a histogram. The results demonstrate a modest improvement compared to the static gain. The RL algorithm also adapts to changing imaging conditions, which can be a tempting advantage in real telescope environments.

4. FUTURE DEVELOPMENT

A key ingredient of the FitAO concept is its open-source development and the authors welcome contributions from other interested AO researchers. Work on several minor improvements are ongoing or planned regarding the functionality and documentation of FitAO. Let us elaborate on two long-term goals.

First, ideas from reinforcement learning have had recent success in AO and data-driven methods seem to hold a great promise for control in real telescope environments. We envision building a library of AO control algorithms openly available on FitAO that enables careful comparison between data-driven methods while promoting the development of new hybrid methods.

Second, our goal with FitAO is to promote open benchmarking of AO algorithms. We believe that a possible step towards this is a platform such as FitAO, which is independent of end-to-end AO simulation environments. Also, this potentially lowers the bar for experts in other fields to contribute to the algorithmic development in AO.

5. CONCLUSIONS

FitAO is a concept platform designed to enable algorithmic development on multiple end-to-end AO simulation environments. Moreover, it is configured to utilize interface specifications of the OpenAI Gym to allow integration of modern control algorithms in the Gym library. We reviewed the functionality and design of FitAO and demonstrated its capabilities with a simple tutorial on applying reinforcement learning to the classical integrator control in a closed-loop SCAO system.

ACKNOWLEDGMENTS

JN was supported by the ESO. JL acknowledges the support from the Emil Aaltonen Foundation. The work of TH was supported by the the Academy of Finland (decisions 320082 and 326961).

REFERENCES

[1] Ferreira, F., Gratadour, D., Sevin, A., and Doucet, N., "Compass: an efficient gpu-based simulation software for adaptive optics systems," in [*2018 International Conference on High Performance Computing & Simulation (HPCS)*], 180–187, IEEE (2018).

[2] Basden, A., Butterley, T., Myers, R., and Wilson, R., "Durham extremely large telescope adaptive optics simulation platform," *Applied optics* **46**(7), 1089–1098 (2007).

[3] Reeves, A., "Soapy: an adaptive optics simulation written purely in python for rapid concept development," in [*Adaptive Optics Systems V*], **9909**, 2173–2183, SPIE (2016).

[4] Conan, R. and Correia, C., "Object-oriented matlab adaptive optics toolbox," in [*Adaptive optics systems IV*], **9148**, 2066–2082, SPIE (2014).

[5] Auzinger, G., *New Reconstruction Approaches in Adaptive Optics for Extremely Large Telescopes/submitted by Dipl.-Ing. Günter Auzinger*, PhD thesis, Universität Linz (2017).

[6] Rigaut, F. and Van Dam, M., "Simulating astronomical adaptive optics systems using yao," in [*Third AO4ELT Conference-Adaptive Optics for Extremely Large Telescopes*], **136** (2013).

[7] Carbillet, M., Vérinaud, C., Femenía, B., Riccardi, A., and Fini, L., "Modelling astronomical adaptive optics-i. the software package caos," *Monthly Notices of the Royal Astronomical Society* **356**(4), 1263–1275 (2005).

[8] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W., "Openai gym," *arXiv preprint arXiv:1606.01540* (2016).

[9] Nousiainen, J., Rajani, C., Kasper, M., and Helin, T., "Adaptive optics control using model-based reinforcement learning," *Optics Express* **29**(10), 15327–15344 (2021).

[10] Nousiainen, J., Rajani, C., Kasper, M., Helin, T., Haffert, S., Vérinaud, C., Males, J., Van Gorkom, K., Close, L., Long, J., et al., "Towards on-sky adaptive optics control using reinforcement learning," *to appear in Astronomy & Astrophysics, arXiv preprint arXiv:2205.07554* (2022).

[11] Nousiainen, J., Engler, B., Kasper, M., Heritier, C. T., Rajani, C., and Helin, T., "Advances in model-based reinforcement learning for adaptive optics control," in [*Adaptive Optics Systems VIII*], SPIE (2022).

[12] "Wrappers - gym documentation." https://www.gymlibrary.ml/content/wrappers/.

[13] "Environment creation - gym documentation." https://www.gymlibrary.ml/content/environment_creation/.

[14] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347* (2017).

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

1039. LIPIÄINEN, SATU. The role of the forest industry in mitigating global change: towards energy efficient and low-carbon operation. 2022. Diss.

1040. AFKHAMI, SHAHRIAR. Laser powder-bed fusion of steels: case studies on microstructures, mechanical properties, and notch-load interactions. 2022. Diss.

1041. SHEVELEVA, NADEZHDA. NMR studies of functionalized peptide dendrimers. 2022. Diss.

1042. SOUSA DE SENA, ARTHUR. Intelligent reflecting surfaces and advanced multiple access techniques for multi-antenna wireless communication systems. 2022. Diss.

1043. MOLINARI, ANDREA. Integration between eLearning platforms and information systems: a new generation of tools for virtual communities. 2022. Diss.

1044. AGHAJANIAN, SOHEIL. Reactive crystallisation studies of CaCO3 processing via a CO2 capture process: real-time crystallisation monitoring, fault detection, and hydrodynamic modelling. 2022. Diss.

1045. RYYNÄNEN, MARKO. A forecasting model of packaging costs: case plain packaging. 2022. Diss.

1046. MAILAGAHA KUMBURE, MAHINDA. Novel fuzzy k-nearest neighbor methods for effective classification and regression. 2022. Diss.

1047. RUMKY, JANNATUL. Valorization of sludge materials after chemical and electrochemical treatment. 2022. Diss.

1048. KARJUNEN, HANNU. Analysis and design of carbon dioxide utilization systems and infrastructures. 2022. Diss.

1049. VEHMAANPERÄ, PAULA. Dissolution of magnetite and hematite in acid mixtures. 2022. Diss.

1050. GOLOVLEVA, MARIA. Numerical simulations of defect modeling in semiconductor radiation detectors. 2022. Diss.

1051. TREVES, LUKE. A connected future: The influence of the Internet of Things on business models and their innovation. 2022. Diss.

1052. TSERING, TENZIN. Research advancements and future needs of microplastic analytics: microplastics in the shore sediment of the freshwater sources of the Indian Himalaya. 2022. Diss.

1053. HOSEINPUR, FARHOOD. Towards security and resource efficiency in fog computing networks. 2022. Diss.

1054. MAKSIMOV, PAVEL. Methanol synthesis via CO2 hydrogenation in a periodically operated multifunctional reactor. 2022. Diss.

1055. LIPIÄINEN, KALLE. Fatigue performance and the effect of manufacturing quality on uncoated and hot-dip galvanized ultra-high-strength steel laser cut edges. 2022. Diss.

1056. MONTONEN, JAN-HENRI. Modeling and system analysis of electrically driven mechatronic systems. 2022. Diss.

1057. HAVUKAINEN, MINNA. Global climate as a commons — from decision making to climate actions in least developed countries. 2022. Diss.

1058. KHAN, MUSHAROF. Environmental impacts of the utilisation of challenging plastic-containing waste. 2022. Diss.

1059. RINTALA, VILLE. Coupling Monte Carlo neutronics with thermal hydraulics and fuel thermo-mechanics. 2022. Diss.

1060. LÄHDEAHO, OSKARI. Competitiveness through sustainability: Drivers for logistics industry transformation. 2022. Diss.

1061. ESKOLA, ROOPE. Value creation in manufacturing industry based on the simulation. 2022. Diss.

1062. MAKARAVA, IRYNA. Electrochemical recovery of rare-earth elements from NdFeB magnets. 2022. Diss.

1063. LUHAS, JUKKA. The interconnections of lock-in mechanisms in the forest-based bioeconomy transition towards sustainability. 2022. Diss.

1064. QIN, GUODONG. Research on key technologies of snake arm maintainers in extreme environments. 2022. Diss.

1065. TAMMINEN, JUSSI. Fast contact copper extraction. 2022. Diss.

1066. JANTUNEN, NIKLAS. Development of liquid–liquid extraction processes for concentrated hydrometallurgical solutions. 2023. Diss.

1067. GULAGI, ASHISH. South Asia's Energy [R]evolution – Transition towards defossilised power systems by 2050 with special focus on India. 2023. Diss.

1068. OBREZKOV LEONID. Development of continuum beam elements for the Achilles tendon modeling. 2023. Diss.

1069. KASEVA, JANNE. Assessing the climate resilience of plant-soil systems through response diversity. 2023. Diss.

1070. HYNNINEN, TIMO. Development directions in software testing and quality assurance. 2023. Diss.

1071. AGHAHOSSEINI, ARMAN. Analyses and comparison of energy systems and scenarios for carbon neutrality - Focus on the Americas, the MENA region, and the role of geo-technologies. 2023. Diss.

1072. LAKANEN, LAURA. Developing handprints to enhance the environmental performance of other actors. 2023. Diss.

1073. ABRAMENKO, VALERII. Synchronous reluctance motor with an axially laminated anisotropic rotor in high-speed applications. 2023. Diss.

1074. GUTIERREZ ROJAS, DANIEL. Anomaly detection in cyber-physical applications. 2023. Diss.

1075. ESANOV, BAKHTIYOR. Adaptive user-controlled personalization for virtual journey applications. 2023. Diss.

1076. SILTANEN, JUKKA. Laser and hybrid welding of high-strength structural steels. 2023. Diss.

LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT

LUT University

ISBN 978-952-335-939-0
ISBN 978-952-335-940-6 (PDF)
ISSN 1456-4491 (Print)
ISSN 2814-5518 (Online)
Lappeenranta 2023