REINFORCEMENT LEARNING IN MULTI-MIRROR ADAPTIVE OPTICS

Lappeenranta-Lahti University of Technology LUT

Master's Program in Computational Engineering, Master's Thesis

2022

Tomi Krokberg

Examiner: Professor Lasse Lensu
Assoc. Prof. Tapio Helin

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Computational Engineering

Tomi Krokberg

Reinforcement learning in multi-mirror adaptive optics

Master's thesis
2022
45 pages, 19 figures, 1 table
Examiners: Professor Lasse Lensu and Assoc. Prof. Tapio Helin

Keywords: adaptive optics, reinforcement learning, convolutional neural networks

When imaging astronomical objects from the earth, the turbulent air in the atmosphere causes perturbations to the wavefront of the arriving light. This can then be seen as a blur in the final images. These perturbations can be minimised by using an adaptive optics system, where they are corrected in real time by using special deformable mirrors. These systems are crucial in exoplanet imaging, where the imaged object can be right next to an object a billion times brighter. The light from this nearby object is blocked using an instrument called a coronagraph. However, any perturbations left at the arriving wavefront cause the light from this brighter object to partly miss the block, causing it to leak into the final image and possibly washing out the planet's light. This leads to a situation where the performance of the adaptive optics system is the limiting factor in the imaging quality, with control algorithms playing a major role. While traditional control algorithms have proven to be quite effective in minimising these perturbations, with the rise of popularity in data-based learning methods, the interests have been shifting towards machine learning. Especially reinforcement learning has been an interesting subject of research, as it only requires a criterion of optimality for the presented solution to be known, rather than the actual solution required by supervised learning methods. This property allows the algorithm to explore and discover optimal control strategies by itself. In this thesis, a reinforcement learning based control algorithm is implemented on a dual mirror adaptive optics system designed for exoplanet imaging. It is also shown to outperform an optimised traditional integrator controller under tested conditions.

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Engineering Science
Laskennallinen tekniikka

Tomi Krokberg

Vahvistusoppiminen useamman peilin adaptiivisessa optiikassa

Diplomityö
2022
45 sivua, 19 kuvaa, 1 taulukko
Tarkastajat: Professori Lasse Lensu ja Apulaisprofessori Tapio Helin

Hakusanat: adaptiivinen optiikka, vahvistusoppiminen, konvoluutioneuroverkko
Keywords: adaptive optics, reinforcement learning, convolutional neural networks

Maanpinnalta tapahtuvassa taivaankappaleiden kuvaamisessa turbulenttinen ilmakehä aiheuttaa vääristymiä saapuvan valon aaltorintamiin. Tämä näkyy kuvissa kohteiden sumentumisena. Tätä ongelmaa voidaan korjata reaaliajassa käyttämällä adaptiivista optiikkaa, joka hyödyntää muotoiltavia peilejä vääristymien korjaamiseen. Nämä ratkaisut ovat erityisen tärkeitä eksoplaneetoiden kuvaamisessa, joissa planeetta sijaitsee usein jopa miljardi kertaa kirkkaamman tähden vieressä. Tämän kirkkaamman tähden valo voidaan estää käyttämällä koronagraafia. Silti, pienetkin ilmakehän aiheuttamat vääristymät johtavat siihen että osa kirkkaamman tähden valosta ohittaa tämän esteen, jolloin eksoplaneetan valo voi peittyä kuvassa tämän alle. Tästä johtuen adaptiivisen optiikan suorituskyky onkin usein kuvanlaadun rajoittava tekijä, jossa käytetyt kontrollimenetelmät ovat merkittävässä roolissa. Vaikka perinteiset kontrollimenetelmät ovat osoittaneet hyviä tuloksia, on huomio viime aikoina keskittynyt datapohjaisiin koneoppimismenetelmiin. Erityisesti vahvistusoppimismentelmät ovat kiinnittäneet huomiota, sillä niiden ohjaamiseen tarvitsee arvioida vain lopputuloksen hyvyyttä, ilman että valmista ratkaisua tarvitsisi tietää. Tämä tarkoittaa että ne voivat itse tutkia ja oppia optimaalisia kontrollistrategioita. Tässä työssä esitellään vahvistusoppimiseen perustuva kontrollialgoritmi, joka on implementoitu eksoplaneetoiden kuvantamiseen suunnitellulle kahden peilin systeemille. Tämän systeemin osoitetaan myös suoriutuvan perinteistä integraattoriohjainta paremmin testatuissa olosuhteissa.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor and examiner, Tapio Helin, for introducing me to the fascinating world of adaptive optics and inverse problems. A big thank you also goes to Jalo Nousiainen, who has helped me with the many problems I have had related to adaptive optics, including guiding me through the practical matters of this thesis. I would also like to thank examiner Lasse Lensu for the invaluable feedback on this thesis and for interesting courses on machine learning.

A special thanks goes to friends and family who have supported me through these studies. The guild of Lateksii also deserves a special mention for all the unforgettable events organised and the ample amounts of help available for almost any problem. Last, but definitely not the least, I would like to thank kurssi-1 gang for the many late nights of varying productivity.

Lappeenranta, August 14, 2022

Tomi Krokberg

LIST OF ABBREVIATIONS

AO	Adaptive optics
CNN	Convolutional Neural Network
COMPASS	COMputing Platform for Adaptive opticS System
DM	deformable mirror
DOF	degrees of freedom
GS	guide star
LReLU	Leaky Rectified Linear Unit
MDP	Markov decision process
MLP	multi-layer perceptron
OOMAO	Object-Oriented Matlab Adaptive Optics
P-WFS	Pyramid wavefront sensor
PI	proportional integral
PSF	point spread function
ReLU	Rectified Linear Unit
RL	reinforcement learning
SH	Shack-Hartmann
WFS	wavefront sensor

CONTENTS

1 INTRODUCTION **7**
1.1 Background 7
1.2 Objectives and delimitations 9
1.3 Structure of the thesis 9

2 ADAPTIVE OPTICS **10**
2.1 Introduction 10
2.2 Wavefront sensing and correction 12
2.3 MagAO-X and coronagraphs 14
2.4 Control algorithms 16
2.5 Simulation and FitAO 17

3 MACHINE LEARNING **19**
3.1 Introduction 19
3.2 Neural networks 19
3.3 Convolutional neural networks 21
3.4 Deep learning 22
3.5 Reinforcement learning 23

4 REINFORCEMENT LEARNING APPROACH FOR CONTROLLING ADAPTIVE OPTICS **26**
4.1 Adaptive optics system calibration 26
4.2 Adaptive optics system control 27
4.3 Dynamics model 27
4.4 Policy model 29

5 EXPERIMENTS **31**
5.1 Setup 31
5.2 Description of experiments 32
5.3 Evaluation criteria 33
5.4 Results 33

6 DISCUSSION **39**
6.1 Current study 39
6.2 Future work 40

7 CONCLUSION **41**

REFERENCES **42**

1 INTRODUCTION

1.1 Background

Optical or near-infrared images of distant stars taken with ground-based telescopes are distorted by turbulence in the atmosphere. Adaptive optics (AO) is a technique developed to improve the imaging quality of the telescope by correcting for the turbulence using a deformable mirror. An example of the importance can be seen in Figure 1 [1]. Although the concept originated from astronomical imaging [2], it has also been used in microscopy to correct the perturbations caused by intracellular fluids [3].

Figure 1. Neptune imaged using Very Large Telescope with and without AO [1].

These systems are often operated using standard proportional integral (PI) controllers in a closed loop, requiring a dedicated calibration procedure. In a closed-loop operation, the error measurements are made after the correction, which means that the residuals of the error are measured. This has the benefit of being able to see the imperfections in the applied correction and iteratively improve the control. With a standard PI controller, a partial correction of the measured residual error is applied at each step. The amount of correction applied is controlled by a parameter called gain. Lower gains minimise the effect of measurement errors and help achieve better control stability, while higher gains allow faster response times. This means that the optimal gain for the system depends on the prevailing conditions in the atmosphere and is a balancing act to maximise system

stability while minimising response time. With the physical and computational delays present in the system, this method of control is also inevitably lagging behind the real state of turbulence. However, since a large part of this turbulence can be assumed to be in frozen flow on the millisecond time scale of AO [4], a significant amount of it can be predicted. The frozen flow states that these turbulent layers can be modelled as static random fields, which are then shifted across the sky at the speed of the wind. This has led to the development of predictive control algorithms that use past telemetry data to predict the true state of the sky at the time of correction, minimising the temporal error caused by the system delays [5]. However, real systems still suffer from dynamic modelling errors such as misregistration, optical gain effect for Pyramid wavefront sensor (P-WFS) and temporal jitter. These errors can require external tuning and recalibration of the system to keep the performance optimal.

This has led to an interest in fully data-driven control algorithms to cope with these problems. In particular, reinforcement learning (RL) methods have been shown to have potential [6, 7]. RL algorithms learn by interacting with an environment and maximizing the reward associated with the actions it chooses in the environment. This means that by designing a successful reward function (giving a value of how good any given action was) and choosing the right type of RL algorithm, one could automate the learning of the control algorithm without accurately knowing the underlying model of the system or how to take optimal actions.

An area where AO systems are especially important is exoplanet imaging. When exoplanets are imaged, they usually reside close to stars up to a billion times brighter. This leads to the light that arrives from the planet being washed out by the light arriving from the star. To imagine these exoplanets, the light coming from the star has to be filtered out. This is done using an instrument called a coronagraph. However, any distortions in the arriving wavefronts cause the stars' light to partly miss this block, possibly overpowering the light of the planet. Thus, the performance of the AO system dictates the amount of light that leaks into the image and, as such, the planets that can be imaged. These systems are often called extreme adaptive optics, for the high levels of performance required from the system. This makes them a prime candidate for control algorithm research.

MagAO-X [8] is one of such extreme adaptive optics systems used to aid in the imaging of exoplanets. It uses two deformable mirrors in series to achieve high spatial resolution in wavefront correction. The first "woofer" mirror allows for higher actuator ranges at lower spatial resolutions, while the following "tweeter" mirror provides higher spatial resolution at lower actuation ranges.

1.2 Objectives and delimitations

In this thesis, a novel RL algorithm for two-stage AO system inspired by [9] is developed. This system consists of two deformable mirrors, where one mirror controls the lower frequencies using a more traditional control algorithm, and the other controls the higher frequencies with RL-based control. The system uses one guide star (GS) and a single wavefront sensor (WFS) to measure the wavefronts. The tests are carried out in the simulator environment COMputing Platform for Adaptive opticS System (COMPASS) [10] using an in-house interface called FitAO mainly developed by the author. To be more precise, the objectives of this thesis are:

- Give an overview on the principles of AO and RL.
- Describe recent (RL) AO control algorithms and their performance based on a literature review.
- Test a model-based RL algorithm inspired by [9] on a AO system with two deformable mirrors in a simulated environment and compare the results with the optimised integrator controller.

1.3 Structure of the thesis

In Chapter 2 the basics of AO and coronagraphs are introduced. A literature review on current RL based control algorithms is also performed. Chapter 3 introduces the relevant background information on machine learning and neural networks. The proposed algorithm is described in Chapter 4. The algorithm is tested with methods explained in Chapter 5, where the results are also shown. Chapter 6 is used to discuss these results and the work in general in more detail. Chapter 7 is used to give the final conclusions.

2 ADAPTIVE OPTICS

2.1 Introduction

Adaptive optics is a system that is used to physically correct perturbations in arriving wavefronts to enhance the imaging quality of the systems. This is achieved by using a deformable mirror (DM) to correct the approaching wavefronts. The perturbations in wavefronts are measured using an WFS and, in the case of astronomical imaging, a secondary light source called GS is used to obtain enough photons to make accurate measurements of the perturbations [11]. These GSs can be bright astronomical objects near the imaged object, or are created using lasers to excite atoms high in the atmosphere to glow. As the light sources are located near the imaged object, the light traveling back from them experiences roughly the same perturbations as the light from the imaged object and as such can be used to estimate the original perturbations. A simplified illustration of the AO system is shown in Figure 2.

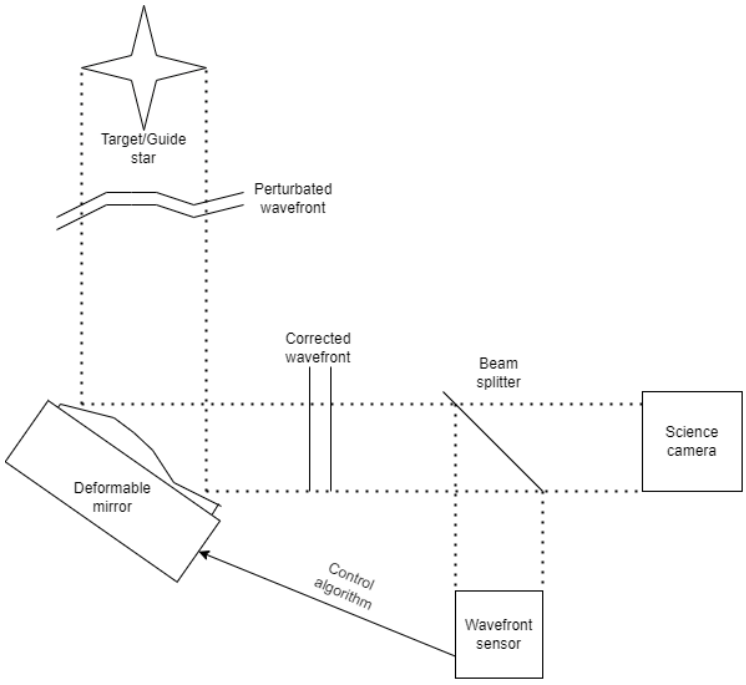
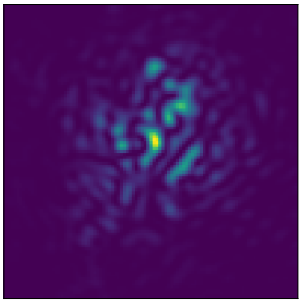


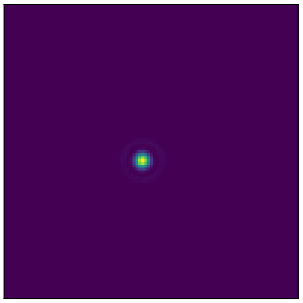
Figure 2. The basic components of an AO system working in a closed-loop.

The visual artefacts that AO tries to correct show in short-exposure images as speckles (example shown in Figure 3). These speckles then average to a blur that can be seen in images without AO. A single sine-wave aberration on the pupil plane will result in two

focal-plane speckles. When using a Fourier transform to approximate any aberration on the pupil plane as a sum of such sine waves, it quickly becomes apparent that even small amounts of aberration left lead to large amounts of speckles.



(a) Uncorrected image with visible speckles.



(b) Diffraction limited image with a very dim airy disk around the star.

Figure 3. Simulated comparison of short-exposure images of a single star under realistic uncorrected conditions and ideal conditions.

From the diffraction limited image in Figure 3 another important concept related to the optical performance of the AO system can be observed. This image includes a feature called an airy disk, a dim ring around the point light source in the middle, caused by the optics of the AO system. In fact, under diffraction limited conditions the optics generate multiple of these airy disks, but most of them are often too dim to be observed in the final image. A one-dimensional representation of an Airy pattern with multiple airy disks can be seen in Figure 4. This response to a point source is called the point spread function (PSF) of the system, and the final image generated by the optical system is a convolution of this PSF and real light sources. In fact, the speckle and diffraction limited images shown in Figure 3 are representations of the PSF for the respective system, since the target imaged in them is a point source. The speckle image in particular showcases how perturbations left in the wavefront affect the PSF.

One common way to evaluate the performance of the AO system is to use a Strehl ratio [11]. It depicts the ratio of true central intensities of the PSF of an ideal diffraction-limited telescope and the one taken from the measurement, so

$$\mathrm{Sr} = \frac{I_{\mathrm{real}}(0,0)}{I_{\mathrm{ideal}}(0,0)} \tag{1}$$

where Sr is the Strehl ratio, $I_{\text{real}}(0,0)$ is the central intensity measured and $I_{\text{ideal}}(0,0)$ is the central intensity of an ideal telescope. Even for smaller aberrations this central intensity can drop significantly [12] and it can not ever exceed the ideal one, meaning the Strehl value will always be between one and zero. As such, it provides an intuitive metric for evaluating the AO system performance.

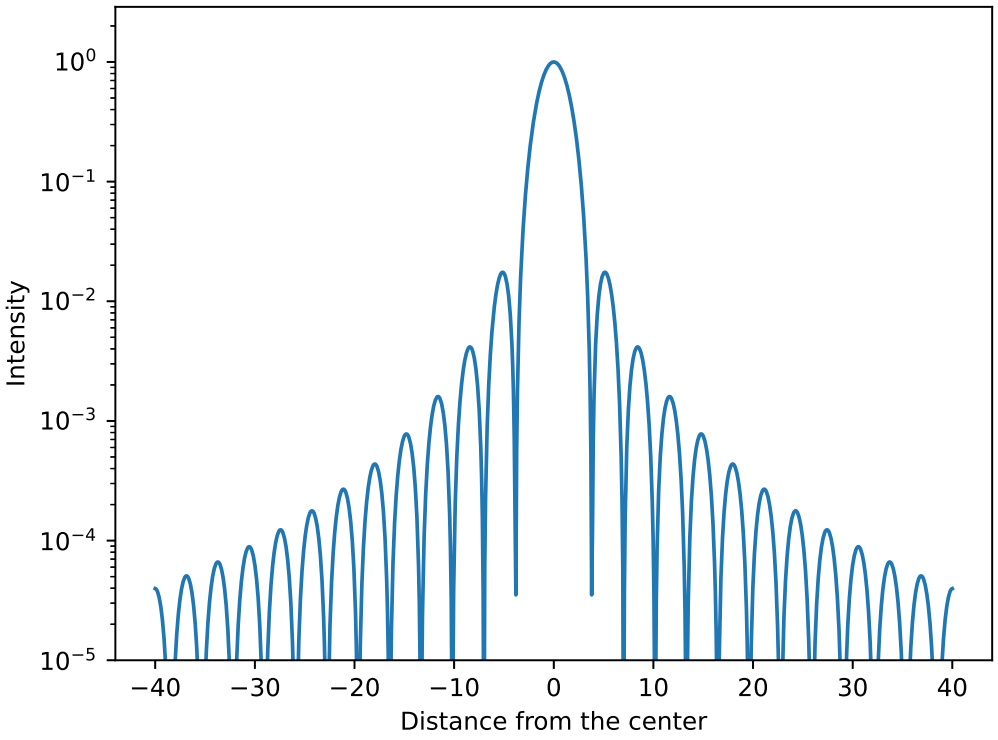


Figure 4. One-dimensional representation of the diffraction limited PSF Airy pattern generated by circular apertures. The central peak shows the intensity of the imaged point source, while each pair of peaks surrounding it correspond to an Airy disk.

2.2 Wavefront sensing and correction

The wavefront sensor is used to measure the spatial shape of the arriving wavefront [11]. If the WFS is located before the DM, the system works in so-called open-loop mode. This means that it directly observes the arriving wavefronts. However, more often, it is more interesting to measure the residual wavefronts, and as such the WFS will be located after the DM. This is called a closed-loop system. Compared to the open-loop system, it has the added benefit of being able to measure also the error in the applied corrections.

There exist multiple types of WFS, but perhaps the two most common are Shack-Hartmann (SH) and Pyramid (P) WFSs. SH is the older and more widely used of these two, but P-WFS has been shown to perform significantly better [13], making it the main focus of current research.

P-WFS works by using a pyramid-shaped prism (often four-sided) on the focal field as a spatial Fourier filter, splitting the incoming wavefront into four (in the case of the four-sided pyramid) intensity fields [14]. These four fields are denoted here as I_1, I_2, I_3 and I_4 and visualized in Figure 5. As the light takes a slightly different path for each of these images, it is possible to calculate so-called slopes from them that positively correlate with the actual gradients in the incoming wavefront. The slopes can be estimated using

$$S_x(x,y) = \frac{(I_1(x,y) + I_2(x,y)) - (I_3(x,y) + I_4(x,y))}{I_0} \tag{2}$$

$$S_y(x,y) = \frac{(I_1(x,y) + I_4(x,y)) - (I_2(x,y) + I_3(x,y))}{I_0} \tag{3}$$

where $I_i(x,y)$ is the intensity in the sub-aperture located at (x,y) in the quadrant i integrated over a modulation cycle and I_0 is the average intensity per sub-aperture of the incoming beam [15].

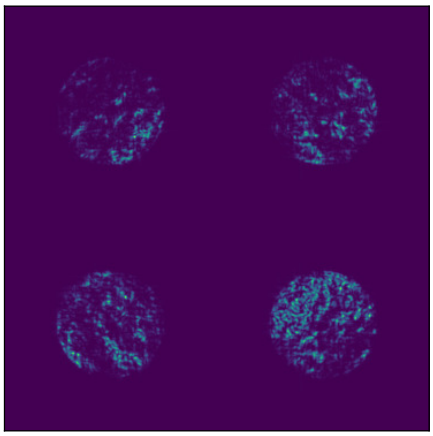


Figure 5. Simulated P-WFS image showing the four intensity fields.

The P-WFS has a important parameter called the modulation radius, which determines the sensitivity and dynamic range of the sensor [16]. Higher amounts of modulation increase the dynamic range of the sensor, allowing it to measure larger aberrations. This can be a

useful feature when starting to close the loop, ie. stabilize the control loop, as the scale of the aberrations is largely determined by the atmosphere. Once the loop is closed and the aberrations left are quite small, decreasing the amount of modulation increase the sensitivity of the sensor and thus enabling more accurate measurements.

Once the wavefronts have been measured, deformable mirrors actively change their shape to keep the residual perturbations at a minimum. The mirrors consist of many actuators, which can be individually controlled to achieve the required corrections. Many types of deformable mirrors exist, like the tip-tilt mirror used in the P-WFS for modulation. As the name suggests, it can only affect the tip and tilt of the incoming wavefront. Other types of mirrors consist of a thin reflective sheet, that is deformed by pushing/pulling the surface using actuators. This allows for more localized corrections on the wavefronts. It is not uncommon for systems to have multiple deformable mirrors [11], for example, a tip-tilt mirror before the sheet mirror to straighten the wavefront while the sheet mirror handles the rest of the perturbations.

2.3 MagAO-X and coronagraphs

MagAO-X [8] is an experimental coronagraphic extreme AO system designed to use the latest cutting-edge technologies to enable imaging of exoplanets and other high-contrast and/or high-resolution astronomical observations. It is used in the Magellan Clay 6.5-meter telescope. For AO it uses a dual DM system, a lower resolution higher actuation range "woofer" mirror followed by a higher resolution lower actuation range "tweeter" mirror. For wavefront detection a P-WFS is used in closed-loop configuration. The structure of this system is visualised in Figure 6.

To enable the imaging of exoplanets, MagAO-X implements a coronagraph before the science instruments. Coronagraphs are telescopic attachments designed to block out the direct light coming from a star. This is important because when imaged at visible wavelengths, the exoplanets are 10^{-9} to 10^{-10} times dimmer compared to their parent star [17]. This means any light coming from them is completely overpowered by the parent star and so imaging exoplanets without blocking the light of the star is infeasible. Coronagraphs work on the principle that the light coming from the star is arriving straight to the telescope, but the light coming from the exoplanet arrives at a slight angle. Then, by placing an object (mask) at the optical axis of the system (where the light is focused to one spot), one can block only the light arriving straight to the system. This can be seen in Figure 7. However, in the presence of atmospheric turbulence the arriving wavefronts are not flat,

and so the light is not focused as neatly causing the light from the star to leak into the final image. Thus, AO is required to flatten the arriving wavefront and minimize this leak.

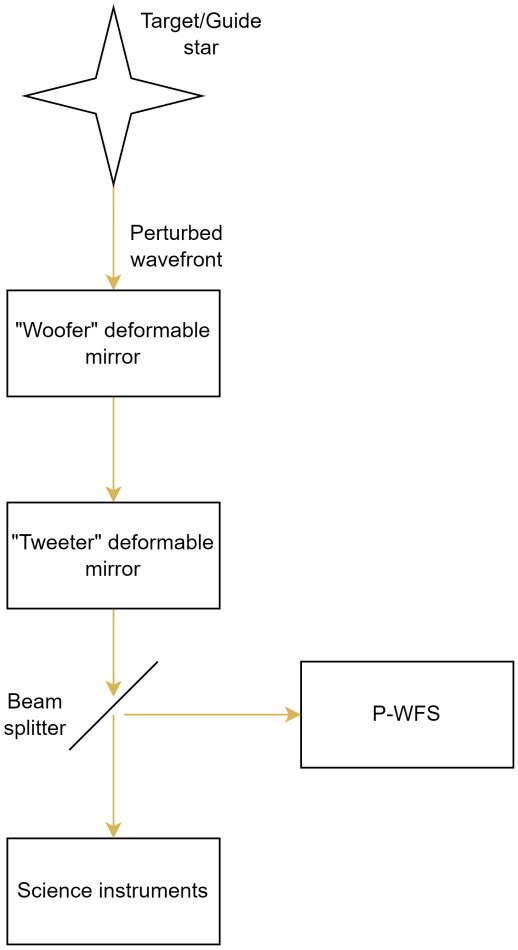


Figure 6. MagAO-X mirror system overview.

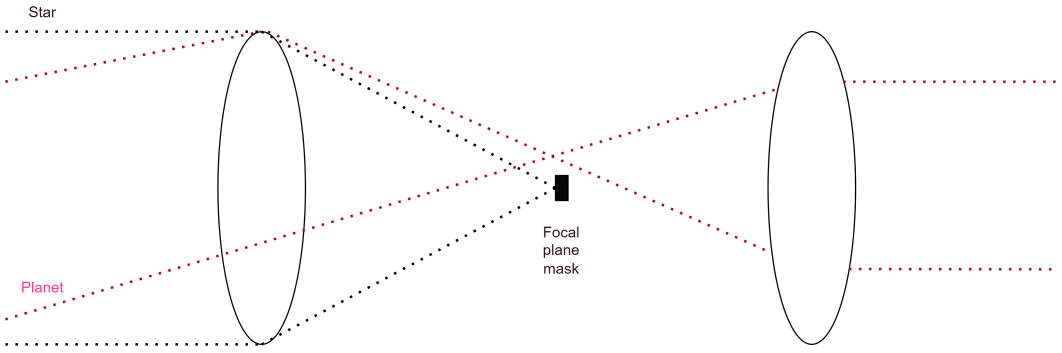


Figure 7. Simplification of the coronagraph system.

2.4 Control algorithms

The requirements for AO control algorithms are strict. To keep up with changing conditions in the atmosphere and effectively correct atmospheric perturbations, the control loops of the AO systems are often run at speeds in the kilohertz range [11]. This means that there are milliseconds at best for measurement, control computation, and commanding the actuators in the DM to move. Also, there are often thousands of actuators to command, which means that being able to compute the control algorithms efficiently is a requirement.

The baseline controller for AO is an integrator controller. For it to work properly, a satisfyingly linear dependence between the WFS measurements and DM commands needs to be assumed [18]. This relation can be shown as

$$w_t = Dv_t + \varepsilon_t \tag{4}$$

where w_t is the wavefront sensor measurement, v_t is the DM command, D is so-called interaction matrix linearly mapping the DM commands to the WFS measurements and ε_t is used to model the measurement noise. Although the interaction matrix could be derived mathematically if accurate enough knowledge of the system is available, it is most often generated experimentally. This happens by poking the DM actuators one at a time (in the WFS linear range) and recording the measured WFS reading. A flat wavefront light source is used during this calibration procedure to capture only the perturbations caused by the actions in the DM. To control the system, an inverse of the interaction matrix D could be used to map the measurements to the DM commands. However, since D is generally ill-conditioned, a regularisation method is needed to invert it. This could be done, for example, by using a truncated singular value decomposition [19]. This resulting pseudo-inverse matrix is commonly called the control matrix, denoted here by C.

Now, the integrator controller can be defined as

$$v_t = v_{t-1} + gC\Delta w_t \tag{5}$$

where v_t is the command given to the DM, v_{t-1} is the previous command given, g is a settable parameter called gain and Δw_t is the residual wavefront. As the initial command can be set to zero and in the closed-loop configuration the residual wavefront being measured, this makes the integrator controller a natural fit for closed-loop operation. The gain parameter g can be used to set how aggressively the system tries to correct to measurements. With a g of one, the mirror is set to exactly what is measured. However, due to

the presence of noise in the system, this is most often not the optimal strategy as there tends to be overshoot in the system, leading to unstable control. As such, values less than one are necessary to achieve stability [18] at the cost of responsiveness. The optimal gain depends a lot on the prevailing conditions and is hard to estimate on the fly, meaning optimal settings are rarely achieved. Also, as the act of inverting the interaction matrix is an ill-posed problem, it is subject to optimization and no single optimal solution exists. This means that better performance could be achieved with alternative control algorithms.

Data-driven control algorithms are currently under research and they have proved to aid the traditional methods to work under a wider range of atmospheric conditions [20]. Furthermore, algorithms based on RL have shown promise. Landman et al. in [6] show a RL based control algorithm to beat traditional integrator controllers in the case of low degrees of freedom (just a tip-tilt mirror to control). Pou et al. in [21] show an RL algorithm correcting the commands made by an integrator controller resulting in higher imaging quality. Nousiainen et al. in [7] beat the integrator controller on a purely RL based controller, with the AO system complexity somewhere in between the work of Landman et al. and Pou et al. Algorithms made by Landman et al. and Pou et al. were so-called model-free RL algorithms, meaning the system dynamics were not learned but just the control task. This resulted in faster inference times but at the cost of requiring more samples to achieve good performance. The model-based approach implemented by Nousiainen et al. required less than 10,000 samples to reach the final performance, while the simpler scenario with a model-free approach required about 40 000 samples, and the more complex several hundred thousand samples to reach the final performance levels. However, the model-based method used by Nousiainen et al. cannot run on larger AO systems due to the slower policy inference. It is also important to note that the testing of these RL algorithms has also been conducted purely on simulations and test benches, so performance on the real telescopes and atmosphere has not yet been seen.

2.5 Simulation and FitAO

Due to the high specificity of developing algorithms directly for telescopes, the control algorithms are often developed and tested on simulators before trials on real hardware. Typically, research groups have their own simulator environments in which to develop and validate their algorithms. Only some of these are publicly available (for example, Object-Oriented Matlab Adaptive Optics (OOMAO) [22] and COMPASS [10]) and all have some differences in how and what things are implemented. This makes the comparison of different control algorithms harder as it would often require rewriting the algorithms from

scratch to the specific environment that the research group uses.

To combat this issue, FitAO was created. It creates a unified interface to control multiple different simulators (currently OOMAO and COMPASS) using a Python interface. It also follows the OpenAI Gym [23] specifications for the interfaces. This means a large library of pre-made RL algorithms that could be tested on the AO control. Using Python as the programming language also means that a large library of scientific, machine learning and data-analytic tools are available for the end-user.

3 MACHINE LEARNING

3.1 Introduction

Machine learning means creating algorithms that can learn patterns from data. This means instead of making a decision based on hard-coded features, machine learning algorithms are given data from which the decision rules are learned. A criterion is the way to control the learning and acts as a final goal for the system. One criterion could be to correctly categorize images into different classes (does this image contain a dog or a cat?) or to predict the temperatures of the following days. Machine learning use cases vary greatly and different techniques have been used, for example, to play a multiplayer video game at the professional level [24], translate text into different languages [25], and build self-driving cars [26].

3.2 Neural networks

A common basis for many modern machine learning methods lies in multi-layer perceptron (MLP). This is due to its property of being a universal approximator [27], which means that it can approximate any finite-dimensional function to an arbitrary degree. The structure of MLPs is modelled after the way the human brain functions and, as such, consists of many simple computational units called artificial neurons. These artificial neurons each perform a simple arithmetic operation of multiplying each input by a given weight and then summing all the inputs together. In addition, a bias is often added to the sum, resulting in an operation

$$\text{output} = w_1 * x_1 + w_2 * x_2 + \cdots + w_n * x_n + b \tag{6}$$

where w_i is the weight corresponding to the i:th input, x_i is the i:th input, n is the number of inputs and b is the bias. After the output is formed, some non-linear operation such as sigmoid

$$S(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

or Rectified Linear Unit (ReLU)

$$R(x) = \max(0, x) \tag{8}$$

is often applied. This non-linear operation is called an activation function, and a non-polynomial one is required for the MLP to function as a universal approximator [28].

Then, multiple of these neurons are connected to the same input features, forming a group called a layer. Each neuron will have and update its weights independently of the other neurons, allowing each to specialise for detecting certain features in its input. Then by connecting outputs of these neutrons to another layer of neurons we have a MLP. Traditionally, these networks are fully connected, meaning that the output of each neuron is connected to every neuron in the next layer. The common high-level structure is to have an input layer, some amount of hidden layers depending on the complexity of the task, and an output layer, from which an example is visualised in Figure 8.

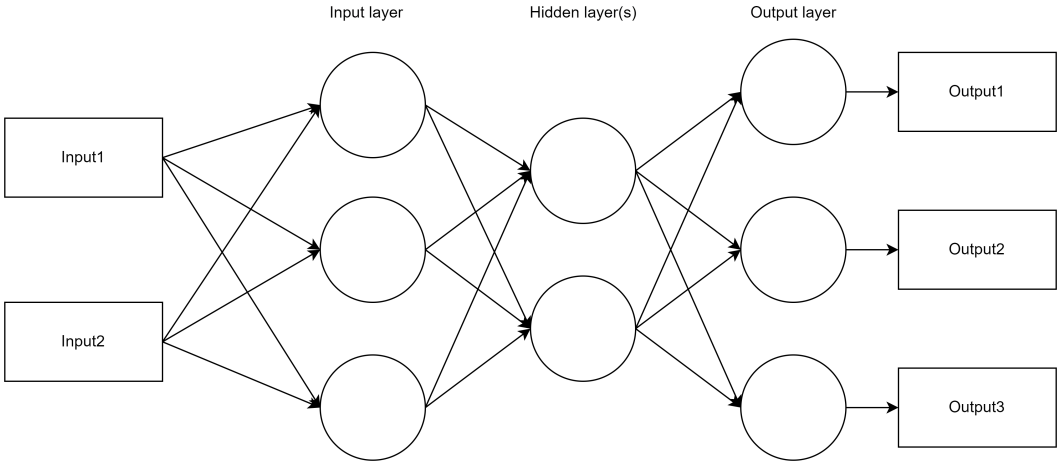


Figure 8. A visualization of an fully connected 3 layer MLP network with 2 dimensional input and 3 dimensional output. Each circle represents an artificial neuron.

For MLPs to approximate a function they need to be trained on examples from the given function. This is done using a process call backpropagation. In this process the weights of the network are typically initialized to be random and then some data is fed trough the network to evaluate its outputs. When the expected output is known (called supervised learning, explored in more detail at the end of Section 3.4), it is possible to calculate the element-wise error. This can be thought of as the gradient of the output, e.g. in which direction our output values should move to result in a correct answer. Using the weights of the previous layer and these error values, it is possible to calculate the errors for the neurons on the previous layer. This process can be repeated until the error is propagated back to the first layer; hence the name backpropagation. Using this error information and information gathered while evaluating the network prediction, it is possible to update the

network weights to match more closely to expected results. However, these errors are not used as is and are scaled down using some value to achieve stability in training. This parameter is often denoted as the learning speed and is subject to optimisation. For a more detailed explanation, see [29].

3.3 Convolutional neural networks

With the advancement of digital cameras and the human dependence on visual information, the use of images as input for machine learning algorithms has become quite common. Although MLPs can use images as input, they require the image to be flattened to a vector, which is often quite large. This, combined with the often fully connected nature of MLPs tends to lead into a large amount of unnecessary connections and, as such, parameters to store/optimize. After all, information in images is often spatial, meaning that the individual pixels' information is mostly relevant compared to the values of the nearby pixels. This information is lost when every neuron makes decisions based on the value of every single pixel and these spatial connections are no longer present. A Convolutional Neural Network (CNN) can be used to solve this problem. By using an convolutional kernel which connects only a small local group of pixels at a time, the spatial information is retained. Intuitively these convolutional kernels can be thought of as filters for an image. Here, the terms filter and convolutional kernel are used interchangeably. By sliding this filter over the image and computing the dot product of the filter and the current part of the image it is possible to generate images called feature maps, which is showcased in Figure 9.

These filters are $n \times m \times d$ tensors of weights, where n and m are often quite small and d is the amount of features in the input data. For example, in the case of RGB images the number of features d is 3, one for each colour channel. Then, for each filter used in a layer, a feature map is generated, and so forth for the next layers. It is important to note that for each layer of filters, n and m can be different, and so it is possible to focus on features of different scales on each layer. Filters also have a parameter called stride, which tells how many pixels the filter slides at each step. Larger stride values enable us to lower the spatial resolution of the image, which is often preferred in the case of larger images. If the original resolution is to be preserved, the original image must be padded with additional data. Possibly the simplest and still a common method is to simply add zeros to the edges of the image. When the padding amount is chosen correctly, the resolution of the produced feature maps matches the original image.

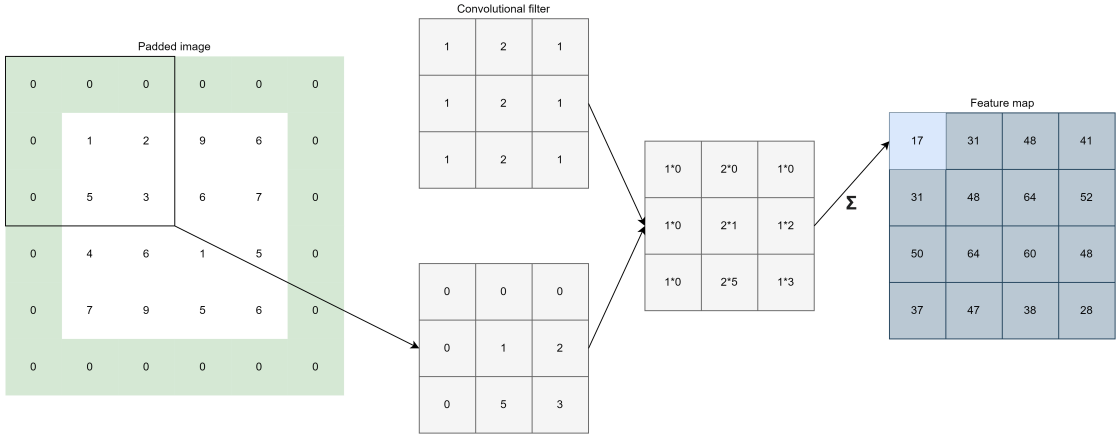


Figure 9. Example of a simple $3 \times 3 \times 1$ convolutional filter and the feature map created by it. The original image is padded with zeros and a stride of one is used to match the original resolution.

3.4 Deep learning

A particularly interesting branch of machine learning is called deep learning. It uses many decision-making layers to divide the problem into easier sub-problems. The first layers can be used to find simple features of the data, for example, simple shapes in images, while the last layers make the final prediction based on the presence of these primary shapes [30]. More layers allow for more stages of abstraction and, as such, allow for representation of more complex structures in the data. The mostly sequential connections between these layers give the "deep" part in the name.

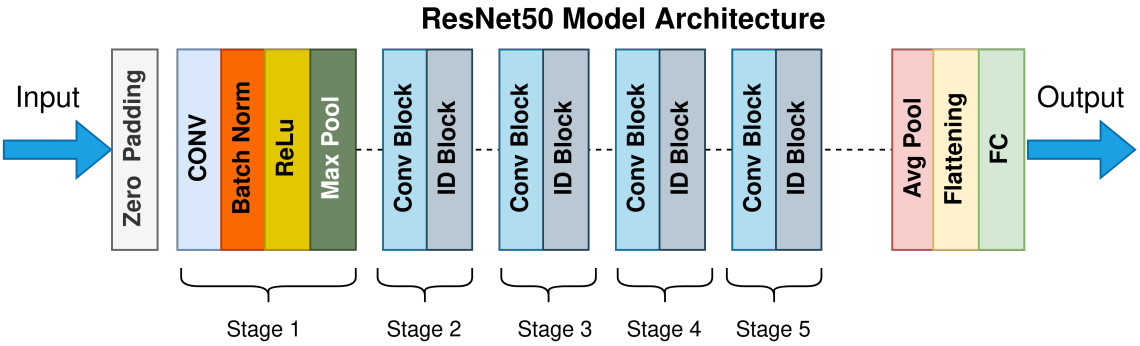


Figure 10. Example of ResNet deep network architecture, combining both convolutional and fully connected layers [31]. Batch norm is used to normalize input data and pooling layers are methods to resize feature maps.

There are multiple different kinds of models where deep learning is often applied like multi-layer perceptrons [32] and convolutional neural networks [33]. Often in deep learning, multiple different methods are also mixed, like in image classification, where the first layers are usually convolutional, while the last few are fully connected layers found in multilayer perceptron networks. This allows one to use the strengths of different kinds of network architecture while also mitigating the shortcomings of any single architecture. An example of such a network structure is shown in Figure 10 [31].

The teaching method can also vary, and the three most common methods are supervised, unsupervised, and reinforcement learning. In supervised learning the expected result is already known and comparing the network results to it can be used as a performance criterion to optimize. This method is often used in classifying data, where the labels already exist, or predicting data, where the optimal result can be known. However, sometimes getting enough classified data to use supervised methods might be unreasonable, and so unsupervised methods can be used. They have no predefined output but try to find patterns inside the data to automatically cluster it. There is no guarantee that the automatic clustering provides the wanted clusters, but if the data is different enough between the cluster satisfactory results can be achieved. The only thing left to do manually is to then label these clusters. The third method of reinforcement learning is a bit of both. Instead of having an optimal solution, it is possible to optimise based on a cost function that only evaluates the goodness of the proposed solution. Then, by trial and error, the algorithm tries to learn how to solve the problem at hand.

3.5 Reinforcement learning

This section is based mainly on a book by A. Zai and B. Brown [30] (unless otherwise noted). Before diving deeper into the world of RL, it is important to introduce the concept of a Markov decision process (MDP). Any control task that has the Markov property is said to be MDP. Markov property states that in any given state, it is possible to know the optimal action to maximize future rewards. In practice, this means that observation of the current state of the world contains all the possible information needed to make the best decision. This is an important property when using RL methods, as many RL algorithms assume that the problem is MDP. For example, when solving static mazes, one look at the maze contains all the information needed to solve the problem, and as such, the problem is a MDP. However, trying to predict the next day's stock price based only on today's price is a futile effort, as some past information would be needed to make good predictions. This means that the problem does not have the Markov property by default. However,

it is possible to define the observation as a set amount of last values (like the last four values of the stock) to artificially introduce the Markov property to the problem. It does not always work though and the final process ends up being a partially observed MDP. This is due to the partially random nature of stock prices, and so it is not possible to have full information. The same problem applies to AO, where the WFS measurements, while mostly predictable, contain some randomness. Luckily the amount of randomness is small enough, that the system can be assumed as a standard MDP [9] and there is no need to worry about the more complicated properties of a partially observed MDP.

Like MDPs, RL algorithms use the same concepts of actions, states, and rewards. The RL algorithm that interacts and learns the environment is often called an actor, and the logic to choose the next action based on the current state is called a policy. For RL the problem is to find an optimal policy

$$\pi_* = \text{argmax}\ \mathbb{E}(R \,|\, \pi) \tag{9}$$

where π_* is the optimal policy given any policy π produces the maximum possible expected reward R based on the current state s and chosen action a. This is done by first randomly sampling the environment (using random actions) and receiving rewards for these actions to enforce or deter from certain behaviors. This information can be used to update the current probabilistic policy π, which in more mathematical terms can be identified with the mapping

$$(A, s) \to \mathbb{P}(A \,|\, s) = \int_A \pi(a \,|\, s) da, \quad s \in S \tag{10}$$

where s is the current state, S is the set of all possible states, and $P(A|s)$ is the probability distribution of the actions A in s. Probability is the likelihood that the given action results in the best possible reward, and, as such, policies can be thought of as a mapping between states and actions. The policy can also be deterministic, in which case it is simply a mapping from the state s to the best estimated action a_*

$$\pi_\theta : \ s \mapsto a_* \tag{11}$$

The optimization of policy π could be done using the raw interactions with the environment or a model of the environment could be used to speed up the process. In so-called model-based RL a model that predicts the next state s_{t+1} based on a action state pair (a_t, s_t) is also trained. Being able to examine the model allows us to compute the entire distribution of states without needing to sample the real environment. As the model can also be trained with supervised learning, it can be more sample efficient than trying

to always sample the real environment (especially if it is slow to sample the real environment). If no model is used, the method is called model-free RL, which might be the optimal solution in cases where learning the dynamics of the environment is difficult.

What makes AO a particularly interesting problem for RL is the size of the action space. For the algorithm to control all the actuators of the systems individually, there would be about 500 to 10000 degrees of freedom (DOF) [9]. This is orders of magnitude larger than the average size of the action spaces in the RL problems. For example the largest action space in collection of benchmark environments created by DeepMind [34] contains 56 DOF and Arcade Learning Environment [35] contains 18 DOF. Also, the observations to the state of the system are indirect, with an ill-posed inverse problem in between.

4 REINFORCEMENT LEARNING APPROACH FOR CONTROLLING ADAPTIVE OPTICS

4.1 Adaptive optics system calibration

First, the interaction matrices are generated separately for both mirrors of the AO system. This is done by pushing and pulling each actuator one at a time and recording the average measured slopes. So,

$$S = \frac{S_{\text{push}} - S_{\text{pull}}}{2} \tag{12}$$

where S is the average slopes, S_{push} is the measured slopes corresponding to pushing the actuator up and S_{pull} is the measured slopes corresponding to driving the actuator down. Each measurement is denoted with $S_i \in \mathbb{R}^{n\times 1}$, where i is the actuator index and n is the number of slope measurements from the system. Then the interaction matrix is

$$D = [S_1, S_2, S_3, \ldots, S_i] \in \mathbb{R}^{n\times i}. \tag{13}$$

This is essentially a linear mapping from the mirror control voltages to the measured slopes. If some other voltage than one unit was used to create the interaction matrix, the matrix needs to be divided by the used voltage. Often, a mapping from slopes to voltages is also needed, and this can be produced using a filtered pseudoinverse of the interaction matrix. SVD [19] is used here to generate this pseudoinverse, often denoted as a control matrix. The modes corresponding to singular values smaller than some portion $n \in [0, 1]$ of the largest singular value are filtered out. With SVD we have

$$D = U \cdot \Sigma \cdot V^\top \tag{14}$$

where U contains the left singular vectors, Σ contains a diagonal matrix of the singular values and V contains the right singular vectors. The pseudoinverse can be then calculated with

$$C = D^+ = V \cdot \hat{\Sigma}^{-1} \cdot U^\top \tag{15}$$

where $\hat{\Sigma}$ is Σ but with the singular values σ_i satisfying the condition

$$\frac{\sigma_i}{\sigma_1} < n \tag{16}$$

set to zero. Here σ_i is the i:th diagonal value of Σ assuming that the diagonal values are in descending order and, as such, σ_1 is the largest singular value.

4.2 Adaptive optics system control

The AO system will be controlled by two independent controllers. The low order "woofer" mirror is controlled by a traditional integrator controller. The higher order "tweeter" mirror located after the woofer is controlled by a slightly modified version of the RL algorithm PO4AO developed by Nousiainen et al. [9].

As a model based RL algorithm the PO4AO algorithm consists of two different parts. The first part is the dynamics model, which tries to learn the underlying dynamics of the system and as such predict upcoming measurements of the system given a history of actions and measurements. The second part is the policy, which can be thought of as the control algorithm. It also takes as an input history of previous observations and actions. Based on these, it estimates the optimal control action to take.

In terms of MDP the AO system is defined as follows. The states s are defined as a set

$$s_t = (\phi_t, \phi_{t-1}, \ldots, \phi_{t-k}, a_{t-1}, a_{t-2}, \ldots, a_{t-(k+1)}) \tag{17}$$

where ϕ_t is the t:th measured wavefront, a_t is the t:th DM command and k denotes the number of past observations used. Both of these are represented in image matrix format which is explored in more detail in Section 4.3. Using this collection of past measurements, it is possible to adequately fulfill the Markov property and learn the RL problem. Both the dynamics and policy model use the same state definition.

4.3 Dynamics model

The dynamics model works as a deterministic mapping from a state action pair (s_t, a_t) to the predicted next wavefront ϕ_{t+1}. Thus, the dynamics model can be written as

$$\hat{p}_\omega(s_t, a_t) = \phi_{t+1}. \tag{18}$$

The dynamics model consists of an ensemble of CNN models with Leaky Rectified Linear Unit (LReLU) activation functions. Using multiple copies of the same algorithm and training each with their own portion of the dataset allows us to avoid the problem of overfitting often present in training dynamics models with low amounts of samples [36]. From this ensemble, an average of all the predicted next states is used as the final prediction.

To be more exact, the model consists of 5 identical architectures with three convolutional layers deep using 3x3 pixel kernels. The first two layers generate 64 feature maps each and are followed by LReLU

$$R(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0.01x & \text{if } x < 0 \end{cases} \tag{19}$$

activation. The final layer generates only one feature map, which is used as the final result for the prediction and, as such, is followed by no activation functions.

The input for the model is a 3D tensor collecting last k measurements and actions into a $n \times n \times (2 * k)$ tensor, where n is the size of the DM actuator grid. Both measurements and actions are presented as actuator images. For past actions the mapping is simple: an image the size of the actuator grid is initialized as zeros and then the value of the pixels is assigned to match the control voltage of the corresponding actuator (see Figure 11). For the measured slopes, we first need to map them to the control voltages of DM. This is done using the control matrix generated during the calibration of the system

$$\text{voltages} = C \cdot s \tag{20}$$

where C is the control matrix and s is the slope vector measured. The same process is then applied to these actuator voltages to generate a similar image. This process can also be easily reversed, that is, mapping the generated images back to command vectors when needed.

The dynamics model is optimised using supervised learning methods. The environment is sampled using a policy to generate actions, from which we can compare the estimated state generated by the dynamics model with the real state generated by the environment. During the "warm-up" period, this policy will be an integrator controller with varied amounts of noise to efficiently teach the dynamics model the relevant information. Later, the RL-based policy will be used to generate actions that are used to teach the dynamics model. The loss function used is a square difference between the expected next state and the true next state, so

$$\text{loss} = \sum_{K} \|\sigma_{t+1} - \hat{p}_{\omega}(s_t, a_t)\|^2 = \sum_{K} \|\sigma_{t+1} - \hat{\sigma}_{t+1}\|^2 \tag{21}$$

where σ_{t+1} is the true next state, $\hat{\sigma}_{t+1}$ is the predicted next state by the dynamics algorithm and K is the collection of samples it is optimized over. The optimization of the back-propagation is done using Adam [37].

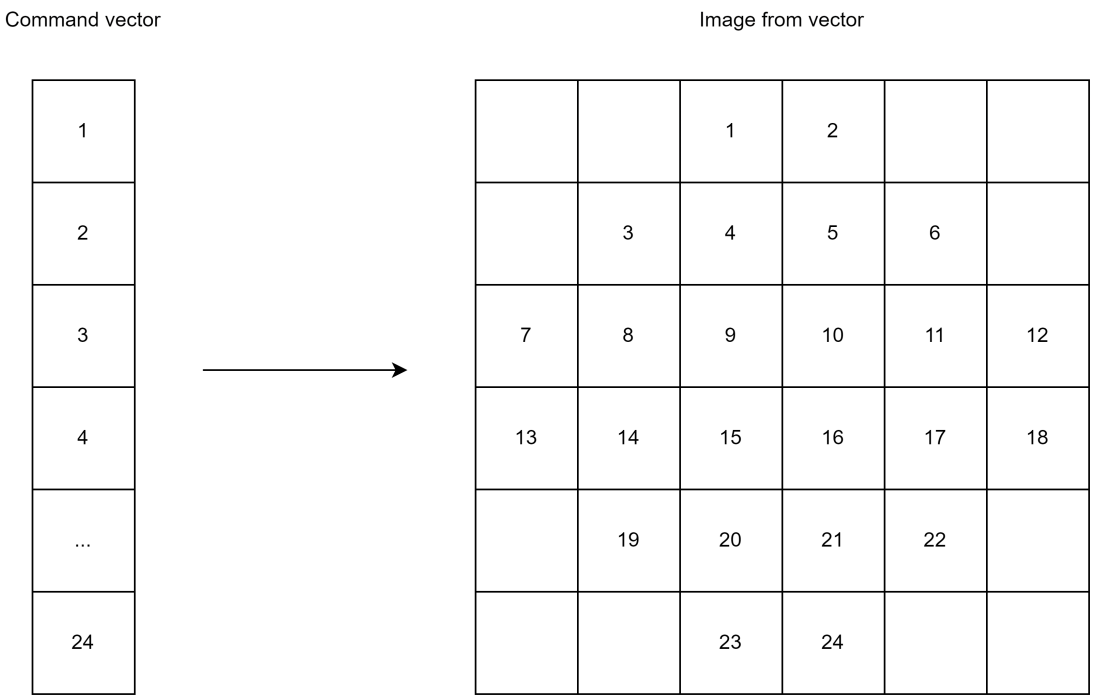


Figure 11. Mapping command vectors to actuator command images. Each number denotes the index of an actuator and those elements contain control voltages, whereas the empty elements contain zeros. The circular shape of the mirror can be observed in the image.

4.4 Policy model

The policy model works as a deterministic mapping from state s_t to the best estimated action a_{t+1}. As such, it can be written as

$$\pi_\theta(s_t) = a_{t+1}. \tag{22}$$

The policy model consists of a single CNN with LReLU activation functions. Like the dynamics model, it consists of three layers with 3x3 pixel kernels, of which the first two generate 64 feature maps and the last produces only one. Similarly, the first two layers have a LReLU activation function and the last layer has no activation function.

Like in the dynamics model, the input of the model is a 3D tensor that collects the last k measurements and actions into a $n \times n \times (2 * k)$ tensor, where n is the size of the DM actuator grid and k is the number of past steps to include. The output is a single actuator image that will be used as an action. To limit the network's ability to learn to control

modes that the WFS does not see (but will still ruin the science image), the output of the algorithm is filtered. This is done using the interaction and control matrices generated during calibration. By first applying the interaction matrix to the output we can map the generated actuator voltages back to slopes which in turn can be returned to filtered actuator voltages using the control matrix. In practise, this is done in a single matrix multiplication using a "filter" matrix

$$F = C \cdot D \tag{23}$$

where C is the control matrix and D is the interaction matrix.

For the optimisation of the policy model, the dynamics model is used to generate estimated observations. As the dynamics model is differentiable, we can backpropagate the evaluated rewards back to the policy model. Due to the delays present in AO systems, we will predict the planning horizon H steps forward, to ensure that the given action is observed in our measurements. In an optimal case where the delay is a known constant, we would match the planning horizon to this delay. However, due to the DM dynamics, temporal jitter and effects of noise exact time for the delay are hard to predict. It follows that the length of planning horizon becomes a balancing act of two effects: two short of a delay and the loop becomes unstable or two long of the delay, and the algorithm tends to overfit. These H steps simulated by the dynamics model are then evaluated using a reward function

$$\hat{r}_\omega(s_t, a_t) = -\|\hat{\sigma}_{t+1}\|^2, \tag{24}$$

which in our case is a negative squared mean of the estimated next state $\hat{\phi}_{t+1}$ calculated using the dynamics model $\hat{p}_\omega(s_t, a_t)$. So, the total reward is

$$\text{reward} = \sum_{s \in K} \sum_{t=1}^{H} \hat{r}_\omega(\tilde{s}_t, \pi_\theta(\tilde{s}_t)), \tag{25}$$

where $\tilde{s}_1 = s$, $\tilde{s}_{t+1} = \hat{r}_\omega(\tilde{s}_t, \pi_\theta(\tilde{s}_t)$ and K is the collection of samples it is optimized over. Like the dynamics model, the Adam algorithm is used to optimise the parameters of the network.

5 EXPERIMENTS

5.1 Setup

The data for the experiment is generated in real time using the COMPASS simulator environment using the FitAO interface. It simulates an experimental MagAO-X coronagraphic extreme adaptive optics system that uses the woofer-tweeter architecture (ALPAO-97 DM as the woofer and Boston Micromachines 2K as the tweeter). As we are using an unmodulated P-WFS, to keep the measurements in the linear range each push and pull action for calibrating the system is 0.01 microns. For the RL algorithm the actuation range of the "tweeter" mirror is limited to ± 0.5 microns. This is done to ensure that the physical limitations of the mirror are not exceeded and to guide the RL algorithms to only correct the residuals of the "woofer" mirror. More detailed simulation parameters can be seen in Table 1.

Table 1. Simulation parameters.

Telescope "MagAO-X"		
Diameter	6.5	meter
Obstruction ratio	14	percent
Sampling frequency	1000	Hz
Active actuators "woofer"	108	···
Active actuators "tweeter"	1822	···
P-WFS subapertures	49×49	apertures
P-WFS modulation	0	λ / D
Photon flux 0/9 mag	1.25×10^8/3.1×10^4	photons / frame / aperture
DM coupling (both)	0.3	percent
DM influence functions	"Gaussian"	···
WFS wavelength	0.85	μm
Science camera wavelength	1.65	μm
Atmosphere parameters		
Fried parameter	16	cm @ 500 nm
Number of layers	3	···
Layer altitudes	0 / 4 / 10	km
C2N	50 / 35 / 15	percent (%)
Wind speeds	10 / 26 / 35	m/s
Wind directions	0 / 45 / 180	degrees
L0 (m)	30 / 30 / 30	m

Table 1. Simulation parameters. (Continued)

PO4AO parameters		
Planning horizon	5	steps
Past DM commands	15	commands
Past WFS measurements	15	frames
CNN ensemble size	5	···
Dynamics iterations / episode	15	steps
Policy iterations / episode	10	steps
Training mini batch size	32	···

5.2 Description of experiments

Two different runs are performed using varying GS magnitudes. The first run is done using a bright guide star of magnitude 0, limiting the amount of measurement error from WFS and, as such, testing the operation of the system under optimal conditions. The second run is done using a GS of magnitude 9, introducing larger amounts of measurement error to WFS and so testing the system under suboptimal conditions.

Each run of the RL method is 100 episodes long, with each episode being 500 frames. At 1000Hz loop rate this means each episode is half a second's worth of data in real time. From these episodes, the first 10 are used for warm-up. During this time both of the mirrors are controlled using separate integrator controllers to collect baseline data for the training. For the "tweeter" mirrors controller, binary noise starting at the amplitude of 0 and ending at the amplitude of 0.025 at the final episode of warm-up is added to show both the dynamics model and the policy model more varied data. This amplitude is increased linearly after each episode. After 10 episodes of warm-up, both the dynamics model and the policy model are trained based on the collected data. In the rest of the episodes, the RL policy model is used to control the "tweeter" mirror. Also, after every episode, both the dynamics and policy model are updated based on the new data gathered. The final episode is then used to calculate the results for the RL algorithm.

For the integrator controller, the system is slightly modified. Only the tweeter mirror is controlled, but unlike in the case of the RL algorithm its actuation range is not limited. This is done to ease the optimisation of the integrator system while still providing the best possible results obtainable by the two mirror system. The integrator controller is

optimised running it at gains from 0.4-1 at 0.05 intervals for 500 frames and calculating the total sum of the Strehl values obtained during the time. The gain with the largest total sum is then used as the final gain for the comparison. In the final benchmarking the integrator is first ran for 500 frames to let it stabilize and after that the final results are computed from the following 500 frames.

5.3 Evaluation criteria

The performance of the system is evaluated on the basis of Strehl values (see Eq. (1)), PSF images, and coronagraph contrast images. For contrast images radial averages are calculated and scaled based on the maximum brightness of the science image. For these contrast images the important factor is the amount of light inside the control radius of the system. This shows as a circle in the contrast images, and everything outside it is not controllable by the AO system. Thus, the goal of the AO system is to minimize the amount of light leaking inside this control radius. The PSF and coronagraph images are presented in logarithmic scale to emphasize any differences between them.

5.4 Results

The PSF image and coronagraph image from the first experiment using the RL based method are presented in Figure 12. The matching images for the integrator controller can be found in Figure 13. Although both PSF images produce clear Airy disks hinting at performance close to diffraction limits, the integrators coronagraph exhibits a wind-driven halo [38] effect, a "butterfly" pattern characteristic of temporal error resulting in a loss of contrast.

The effects of this butterfly pattern can also be seen in Figure 14, where the radial averages of both coronagraphs are shown. The control radius can be observed to be about 75 units, characterised by the bump in the RL radial average. Inside this radius, the RL based method results in better contrast.

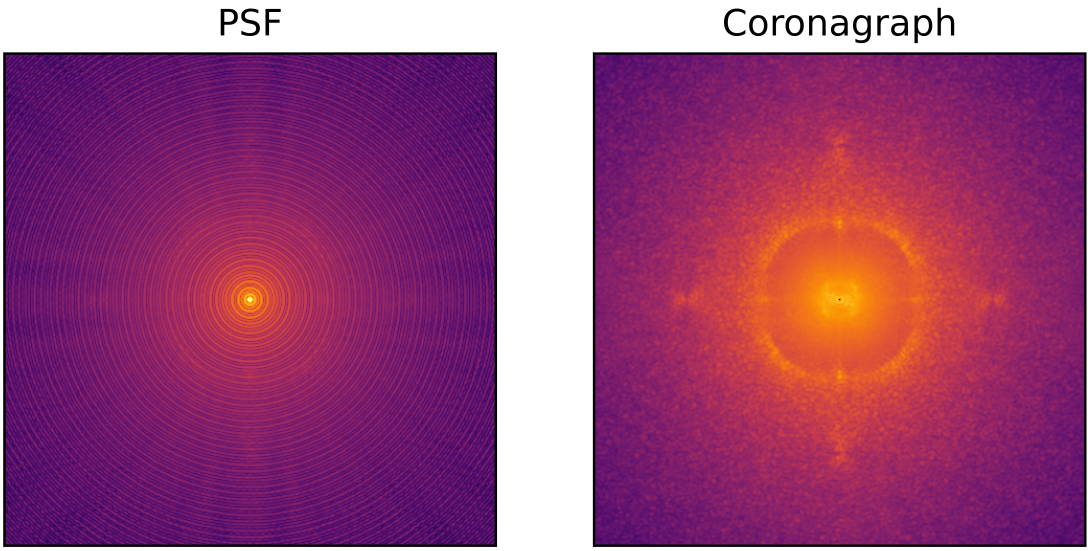


Figure 12. Logarithmic PSF and contrast images of RL based method using a GS of magnitude 0.

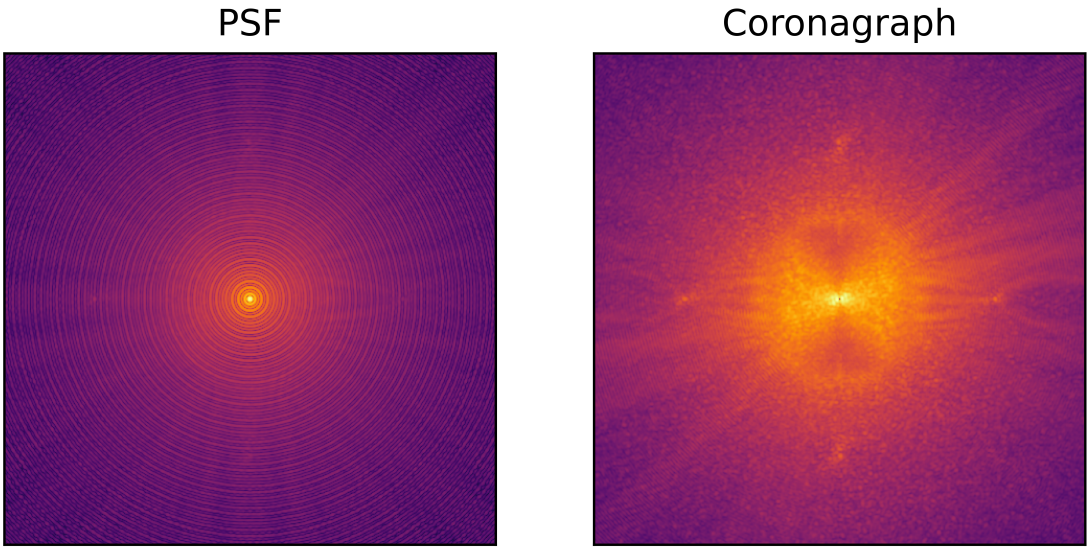


Figure 13. Logarithmic PSF and contrast images of integrator controller using a GS of magnitude 0.

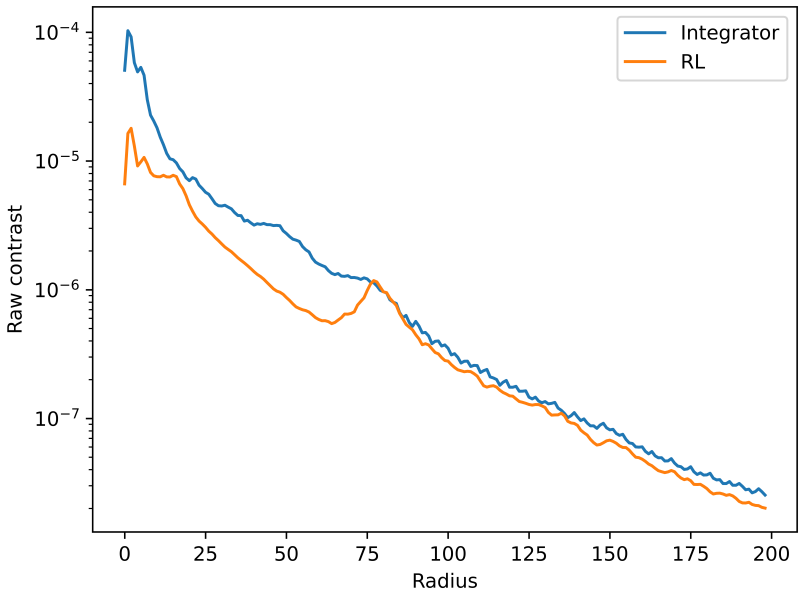


Figure 14. Radial averages of coronagraph images using a GS of magnitude 0.

The training results from the first experiment are shown in Figure 15. The RL based method can be seen consistently beating the integrator controller after about 40 episodes, so 20 000 frames/20 seconds worth of data.

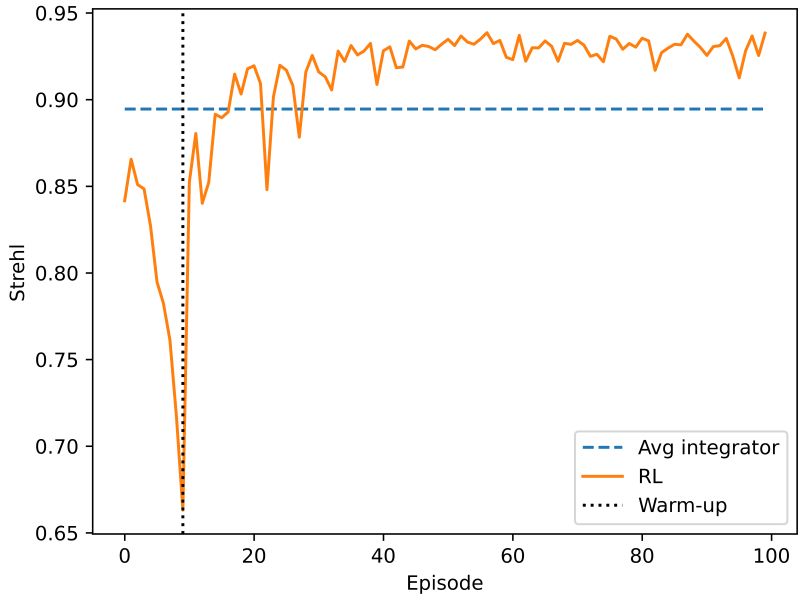


Figure 15. Training progress of the RL model using a GS of magnitude 0.

For the second experiment with the dimmer GS, the PSF and coronagraph images for the RL based method are shown in Figure 16. For the integrator controller, the matching images are shown in Figure 17. Similar patterns compared to the first experiment can be observed. The integrator still exhibits a wind-driven halo in the coronagraph, while the RL based method is more uniform.

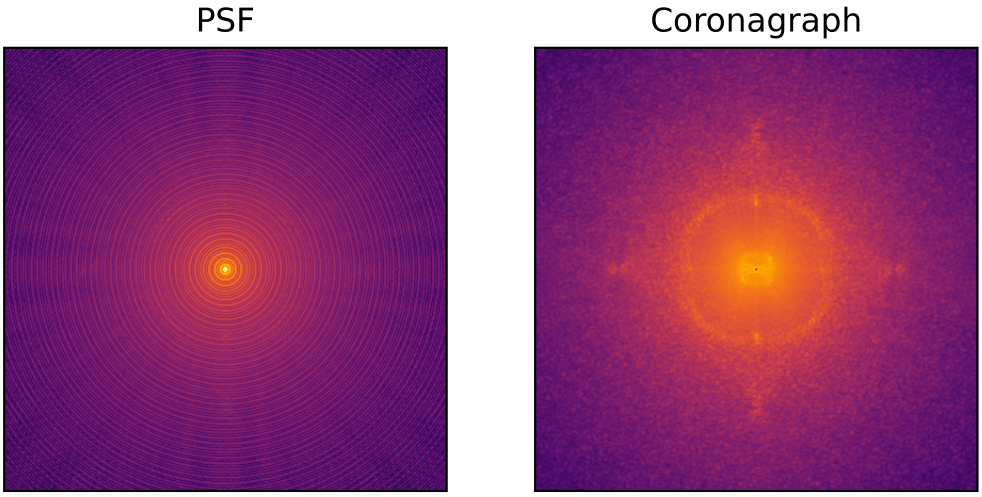


Figure 16. Logarithmic PSF and contrast images of RL based method using a GS of magnitude 9.

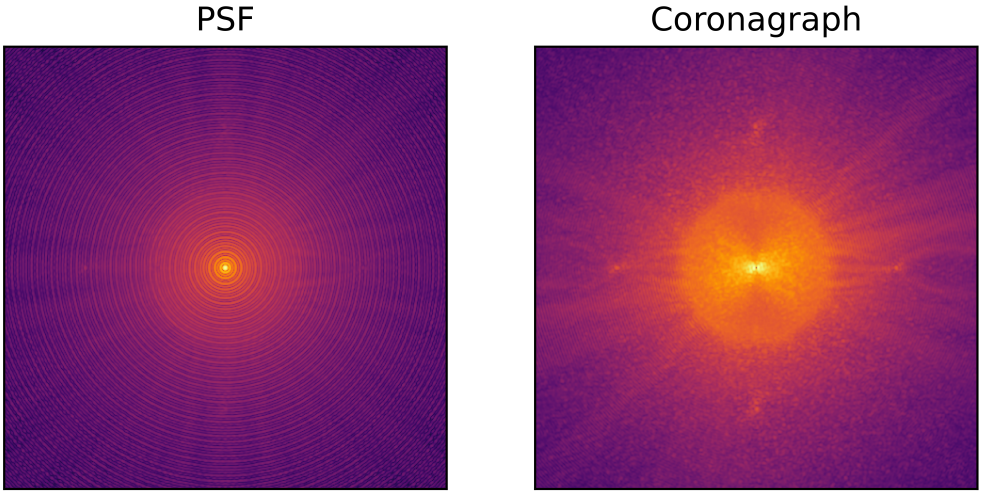


Figure 17. Logarithmic PSF and contrast images of integrator controller using a GS of magnitude 9.

The radial averages for the second experiment are shown in Figure 18. This also follows

the pattern of the first experiment, where inside the control radius the RL based method has improved contrast over the integrator controller.

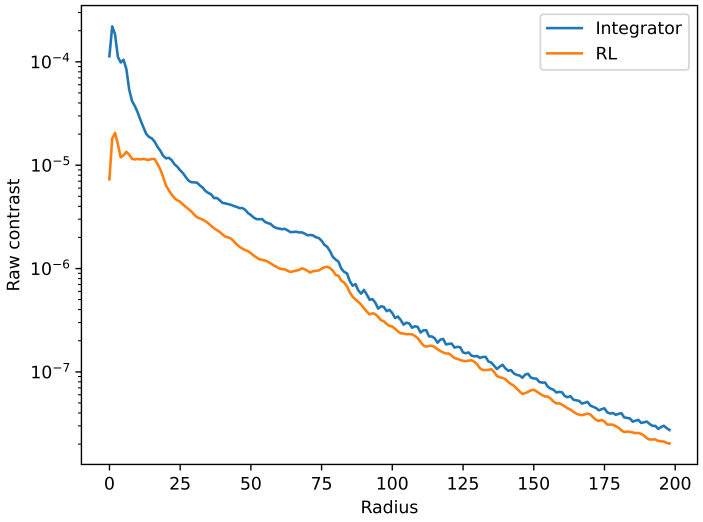


Figure 18. Radial averages of coronagraph images using a GS of magnitude 9.

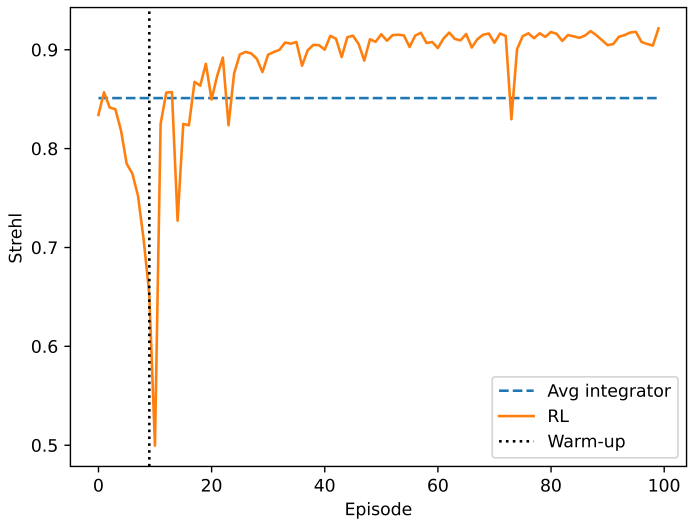


Figure 19. Training progress of the RL model using a GS of magnitude 9.

The training progress for the second experiment is shown in Figure 19. Although in general the RL based method improves the Strehl rate over the integrator controller after

episode 30, one bad episode can be observed between episodes 70 and 80 where the performance drops below integrator levels. This is likely a combination of slight overfitting and differences between used reward function and Strehl values.

6 DISCUSSION

6.1 Current study

The research task of implementing a RL based controller for a multi-mirror AO system was achieved. The presented model was able to outperform the classical integrator controller in every evaluated metric and maintained the fast learning rate of the model-based RL solutions. However, some slow-up in the training process was observed compared to the original PO4AO algorithm. This is likely a combination of two major changes to the original algorithm. The first one is the presence of the integrator controlled "woofer" mirror. The algorithm now had to also learn the effects of this separate controller in order to optimally control the system. Furthermore, since both control methods were independent, nothing stops the woofer mirror from trying to correct any mistakes made by the RL algorithm. In fact, without limiting the actuation range of the second mirror the controllers were observed to use mostly opposite commands, which then cancelled out to achieve the correcting effect. This, of course, was not optimal behaviour, and limiting the actuation range of the second mirror seemed to correct the issue. The second likely reason is a slight change in the system architecture. In the original paper PO4AO used 32 feature maps per convolutional layer. Here this number is increased to 64 feature maps per layer to account for the added complexity of the task. This leads to larger amounts of filters to optimise, and as such longer learning times.

Still, from the experiment results it is clear that the system learnt predictive control and minimised temporal error on the system. The final coronagraph images for the RL based system did not include the "butterfly" artefact characteristic to temporal errors and the contrast was significantly better inside the whole control radius of the system. This increase in performance could also be seen in the Strehl values, which after the initial training were consistently higher than those of the integrator system. With the worse case experiment, one dip in the training accuracy was noticed consistently. The exact cause of this is unclear, but a few possible options exist. It is possible that some overfitting is present in current CNN parameters, as with 32 feature maps this was not observed (but performance in general was slightly worse and not enough tests were ran to be sure). Also, the used reward function (see Eq. (24)) and the Strehl ratio, while generally correlated, are not exactly the same. As such, some artefacts might appear momentarily in the control which are seen in the Strehl ratio, but not immediately in the reward function as the WFS is blind to some wavefront perturbations.

However, while the results in general were positive, the contrast performance achieved in the original PO4AO paper was not matched under similar conditions. This is also likely caused by two main factors. The first one being the more complicated task of controlling a dual mirror system. The other one is the choice of reconstruction method for the control/filter matrices. The original paper used a more physically based Karhunen-Loeve decomposition, which in general is better suited for the AO task. Here SVD was used due to ease of implementation and familiarity with the method, even though it is known to have some undesirable properties [39]. Regardless, the RL based method significantly improved the performance compared to the traditional integrator controller, meaning it can handle even a sub-optimal calibration process.

6.2 Future work

In future work, multiple parts of the RL algorithm could be optimised. Perhaps the most interesting improvement to be made would be to control both of the mirrors using the RL algorithm. This could be achieved with minimal editing to PO4AO by using an orthogonal basis for the commands. This way some amount of the lowest order modes could be controlled by the "woofer" mirror, while the rest of the modes would be controlled on the "tweeter" mirror. The major challenge in this solution is choosing and implementing this basis, and calibrating the AO system using it.

Another possible future improvement would be simply to implement the Karhunen-Loeve basis used in the original PO4AO paper. This should immediately increase the imaging quality to some extent. Other parts of the PO4AO algorithm could also be optimised to squeeze out more performance. For example, the hyperparameters of the network, e.g. learning rate, layer amounts in the neural network, and the number of feature maps per layer and properties of the convolutional kernels in CNN were not rigorously optimised.

7 CONCLUSION

In this thesis, a RL based controller was implemented on a multi-mirror AO system. This system used a traditional integrator controller to control a lower order "woofer" DM and an RL algorithm based on PO4AO algorithm to control the higher order "tweeter" DM. This system was tested under two different conditions. The first experiment used a GS of magnitude 0 to simulate good conditions with minimal measurement errors. The second experiment used a GS of magnitude 9 to simulate more pronounced measurement errors and their effect on the system performance. Under both conditions, the implemented method was shown to improve upon an optimised integrator controller based on the Strehl ratio of the system and the contrast of the coronagraph image.

REFERENCES

[1] P. Weilbacher and ESO. Neptune from the VLT with and without adaptive optics. `https://www.eso.org/public/ireland/images/eso1824b/`, 2018. Accessed: 07-02-2022.

[2] F. Roddier. The Effects of Atmospheric Turbulence in Optical Astronomy. In *Progress in Optics*, pages 281–376. Elsevier, 1981.

[3] M. Schwertner, M. J. Booth, M. A. A. Neil, and T. Wilson. Measurement of specimen-induced aberrations of biological samples using phase stepping interferometry. *Journal of Microscopy*, 213(1):11–19, 2004.

[4] 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. *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, 26(4):833–846, 2009.

[5] O. Guyon and J. Males. Adaptive Optics Predictive Control with Empirical Orthogonal Functions (EOFs). *arXiv:1707.00570 [astro-ph.IM]*, 2017.

[6] R. Landman, S. Haffert, V. Radhakrishnan, and C. Keller. Self-optimizing adaptive optics control with Reinforcement Learning for high-contrast imaging. *arXiv:2108.11332 [astro-ph.IM]*, 2021.

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

[8] J. R. Males, L. M. Close, K. Miller, L. Schatz, D. Doelman, J. Lumbres, F. Snik, A. Rodack, J. Knight, K. Van Gorkom, J. D. Long, A. Hedglen, M. Kautz, Ne. Jovanovic, K. Morzinski, O. Guyon, E. Douglas, K. B. Follette, J. Lozi, C. Bohlman, O. Durney, V. Gasho, P. Hinz, M. Ireland, M. Jean, C. Keller, M. Kenworthy, B. Mazin, J. Noenickx, D. Alfred, K. Perez, A. Sanchez, C. Sauve, A. Weinberger, and A. Conrad. MagAO-X: project status and first laboratory results. *arXiv:1807.04315 [astro-ph.IM]*, 2018.

[9] J. Nousiainen, C. Rajani, M. Kasper, T. Helin, S. Y. Haffert, C. Vérinaud, J. R. Males, K. Van Gorkom, L. M. Close, J. D. Long, A. D. Hedglen, O. Guyon, L. Schatz, M. Kautz, J. Lumbres, A. Rodack, J. M. Knight, and K. Miller. Towards on-sky adaptive optics control using reinforcement learning. *arXiv:2205.07554 [astro-ph.IM]*, 2022.

[10] F. Ferreira, D. Gratadour, A. Sevin, and N. Doucet. COMPASS: An Efficient GPU-based Simulation Software for Adaptive Optics Systems. In *2018 International Conference on High Performance Computing Simulation (HPCS)*, pages 180–187, 2018.

[11] P. Hickson. Atmospheric and adaptive optics. *The Astronomy and Astrophysics Review*, 22(1), 2014.

[12] H. Ottevaere and H. Thienpont. Optical microlenses. In *Encyclopedia of Modern Optics*, pages 21–43. Elsevier, 2005.

[13] T. Yong Chew, R. M. Clare, and R. G. Lane. A comparison of the Shack–Hartmann and pyramid wavefront sensors. *Optics Communications*, 268(2):189–195, 2006.

[14] O. Fauvarque, B. Neichel, T. Fusco, J.-F. Sauvage, and O. Girault. General formalism for fourier-based wave front sensing: application to the pyramid wave front sensors. *Journal of Astronomical Telescopes, Instruments, and Systems*, 3(1):019001, 2017.

[15] C. Vérinaud. On the nature of the measurements provided by a pyramid wave-front sensor. *Optics Communications*, 233(1-3):27–38, 2004.

[16] R. Ragazzoni, E. Diolaiti, and E. Vernet. A pyramid wavefront sensor with no dynamic modulation. *Optics Communications*, 208(1-3):51–60, 2002.

[17] J. Trauger, D. Moody, J. Krist, and B. Gordon. Hybrid lyot coronagraph for WFIRST-AFTA: coronagraph design and performance metrics. *Journal of Astronomical Telescopes, Instruments, and Systems*, 2(1):011013, 2016.

[18] O. Guyon. Extreme adaptive optics. *Annual Review of Astronomy and Astrophysics*, 56(1):315–355, 2018.

[19] C. Hansen. The truncated SVD as a method for regularization. *BIT*, 27(4):534–553, 1987.

[20] R. Landman and S. Y. Haffert. Nonlinear wavefront reconstruction with convolutional neural networks for Fourier-based wavefront sensors. *Optics Express*, 28(11):16644, 2020.

[21] B. Pou, F. Ferreira, E. Quinones, D. Gratadour, and M. Martin. Adaptive optics control with multi-agent model-free reinforcement learning. *Optics Express*, 30(2):2991–3015, 2022.

[22] R. Conan and C. Correia. Object-oriented Matlab adaptive optics toolbox. In *Adaptive Optics Systems IV*, volume 9148, pages 2066 – 2082. International Society for Optics and Photonics, SPIE, 2014.

[23] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv:1606.01540 [cs.LG]*, 2016.

[24] OpenAI, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv:1912.06680 [cs.LG]*, 2019.

[25] M. Xu Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, Y. Wu, and M. Hughes. The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. *arXiv:1804.09849 [cs.CL]*, 2018.

[26] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[27] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[28] M. Leshno, V. Ya. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[30] A. Zai and B. Brown. *Deep reinforcement learning in action*. Manning Publications, 2020.

[31] Gorlapraveen123. ResNet50. `https://commons.wikimedia.org/wiki/File:ResNet50.png`, 2021. Accessed: 07-02-2022.

[32] Z. R. Yang. Multi-layer Perceptron. In *Machine Learning Approaches to Bioinformatics*, pages 133–153. WORLD SCIENTIFIC, 2010.

[33] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-Scale Video Classification with Convolutional Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014.

[34] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller. DeepMind Control Suite. *arXiv:1801.00690 [cs.AI]*, 2018.

[35] M. G. Bellemare, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 2013.

[36] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4759–4770. Curran Associates Inc., 2018.

[37] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs.LG]*, 2014.

[38] F. Cantalloube, O. J. D. Farley, J. Milli, N. Bharmal, W. Brandner, C. Correia, K. Dohlen, Th. Henning, J. Osborn, E. Por, M. Suárez Valles, and A. Vigan. Wind-driven halo in high-contrast images. *Astronomy & Astrophysics*, 638:A98, 2020.

[39] D. T. Gavel. Suppressing anomalous localized waffle behavior in least-squares wavefront reconstructors. In P. L. Wizinowich and D.Bonaccini, editors, *SPIE Proceedings*. SPIE, 2003.