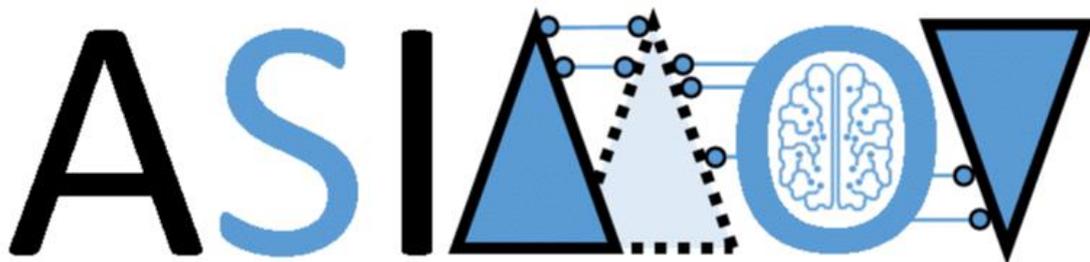


# Proof of Concept Demonstration and Evaluation electron microscopy

**[WP1; T1.2; External Report: D1.2 version 1]**

**public**



**AI training using Simulated Instruments for Machine  
Optimization and Verification**

Version	Status	Date	Page
0	public	2023.03.20	1/34

**PROPRIETARY RIGHTS STATEMENT**

THIS DOCUMENT CONTAINS INFORMATION, WHICH IS PROPRIETARY TO THE ASIMOV CONSORTIUM. NEITHER THIS DOCUMENT NOR THE INFORMATION CONTAINED HEREIN SHALL BE USED, DUPLICATED OR COMMUNICATED BY ANY MEANS TO ANY THIRD

## Document Information

<b>Project</b>	ASIMOV
<b>Grant Agreement No.</b>	20216 ASIMOV - ITEA
<b>Deliverable No.</b>	D1.2
<b>Deliverable No. in WP</b>	WP1; T1.2
<b>Deliverable Title</b>	Proof of Concept – Use case electron microscopy
<b>Dissemination Level</b>	internal
<b>Document Version</b>	Version 1
<b>Date</b>	2022.11.26
<b>Contact</b>	Hans Vanrompay
<b>Organization</b>	THERMO
<b>E-Mail</b>	Hans.Vanrompay@thermofisher.com
<b>Project</b>	ASIMOV
<b>Grant Agreement No.</b>	20216 ASIMOV - ITEA
<b>Deliverable No.</b>	IR1.2
<b>Deliverable No. in WP</b>	WP1; T1.2
<b>Deliverable Title</b>	Specifications and Commonality Analysis
<b>Dissemination Level</b>	external
<b>Document Version</b>	Version M18
<b>Date</b>	2022.11.26
<b>Contact</b>	Hans Vanrompay
<b>Organization</b>	Thermo Fisher Scientific
<b>E-Mail</b>	Hans.Vanrompay@thermofisher.com



The ASIMOV-project was submitted in the Eureka Cluster AI Call 2021

<https://eureka-clusters-ai.eu/>

Version	Status	Date	Page
0	public	2023.03.20	2/34

IR1.2

Proof of Concept Demonstration and Evaluation –  
use case electron microscopy  
public



Version	Status	Date	Page
0	public	2023.03.20	3/34

### Task Team (Contributors to this deliverable)

Name	Partner	E-Mail
Hans Vanrompay	TFS	Hans.Vanrompay@thermofisher.com
Narges Javaheri	TFS	Narges.Javaheri@thermofisher.com
Maurits Diephuis	TFS	Maurits.Diephuis@thermofisher.com
Ilona Armengol	TNO	Ilona.ArmengolThijs@tno.nl
Arjan Mooij	TNO	Arjan.Mooij@tno.nl
Richard Doornbos	TNO	Richard.Doornbos@tno.nl
Jilles van Hulst	TU/e	J.S.V.Hulst@tue.nl
Roy van Zuijlen	TU/e	R.A.C.Zuijlen@tue.nl

### Formal Reviewers

Version	Date	Reviewer

### Change History

Version	Date	Reason for Change

Version	Status	Date	Page
0	public	2023.03.20	4/34

--	--	--

Version	Status	Date	Page
0	public	2023.03.20	5/34

## Abstract

In this document we will describe the outcome of the first Proof of Concept demonstration for the electron microscopy use case. This use case involves the automated correction of defocus and A1 astigmatism at the electron microscope through a reinforcement learning agent. The agent itself has been trained on simulated data, obtained from the digital twin of the electron microscope. In what follows we will first provide the required information on electron microscopy, digital twinning, classical alignment strategies and reinforcement learning.

In the first section we will introduce the electron microscope and discuss the Ronchigram image formation. Next, we discuss the implementation of the Digital Twin and how it enables Ronchigram simulation.

Based on the simulated data we will first present state-of-the classic optimisation strategies, capable of retrieving the optimal alignment. Afterwards, we introduce reinforcement learning and explore how it can be used for minimizing defocus and A1 astigmatism.

Important concepts are first introduced via a the 'ASIMOV-in-a-nutshell' demonstrator. Later on they are applied on both simulated as experimental microscope data.

Finally we discuss the live application at the electron microscope and present an outlook on how to productize this use-case to a product feature.

Version	Status	Date	Page
0	public	2023.03.20	6/34

- 1. Electron Microscopy ..... 10**
  - 1.1 *Transmission Electron Microscope .....10*
  - 1.2 *Ronchigram .....11*
- 2. Digital Twin ..... 13**
- 3. Classical optimization for aberration correction ..... 16**
- 4. Reinforcement Learning ..... 21**
  - 4.1 *Introduction.....21*
  - 4.2 *ASIMOV-in-a-Nutshell demonstrator .....22*
    - 4.2.1 *System to be optimized.....22*
    - 4.2.2 *Experimental set-up .....23*
    - 4.2.3 *Agent behavior .....24*
    - 4.2.4 *Early exploration ..... Error! Bookmark not defined.*
- 5. Reinforcement Learning for aberration correction ..... 27**
  - 5.1 *Architecture ..... Error! Bookmark not defined.*
  - 5.2 *Neural Network for RL..... Error! Bookmark not defined.*
  - 5.3 *Reward function ..... Error! Bookmark not defined.*
  - 5.4 *Data sampling method..... Error! Bookmark not defined.*
  - 5.5 *Outcome..... Error! Bookmark not defined.*
- 6. Live application at the electron microscope..... 29**

Version	Status	Date	Page
0	public	2023.03.20	7/34



6.1 *Interfaces*..... **Error! Bookmark not defined.**

6.2 *Discussion and Results*.....29

6.3 *Conclusion* .....31

**7. Outlook: from use case to product feature** .....Error! Bookmark not defined.

7.1 *Business perspective*..... **Error! Bookmark not defined.**

7.2 *Operator perspective*..... **Error! Bookmark not defined.**

7.3 *Organizational perspective*..... **Error! Bookmark not defined.**

7.4 *Technical perspective* ..... **Error! Bookmark not defined.**

**8. Terms, Abbreviations and Definitions**..... **33**

**9. Bibliography** ..... **34**

Version	Status	Date	Page
0	public	2023.03.20	8/34

IR1.2

Proof of Concept Demonstration and Evaluation –  
use case electron microscopy  
public



Version	Status	Date	Page
0	public	2023.03.20	9/34

## 1. Electron Microscopy

### 1.1 Transmission Electron Microscope

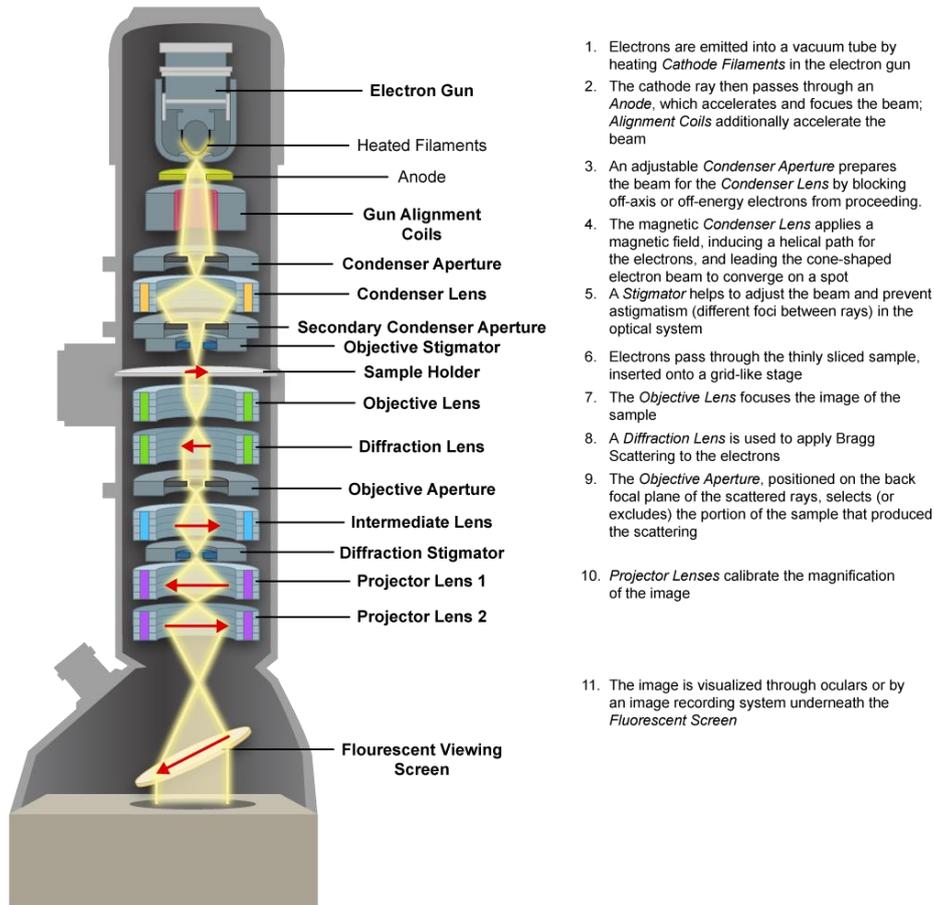
The image formation process in Transmission Electron Microscopy (TEM) is very similar to that of an optical light microscope. The main difference is related to the use of electrons as the source of light and the associated replacement of glass lenses by electromagnetic coils. In TEM, electrons are accelerated towards the specimen using high voltages in order to obtain images with a resolution higher than achievable in optical light microscopes. These electrons are emitted from a thermionic gun or a field emission gun (FEG). Afterwards they pass through a system of condenser lenses, to produce a beam with desired size, intensity and convergence. In TEM mode, a parallel coherent beam is formed, which illuminates the sample uniformly. In Scanning Transmission Electron Microscopy (STEM) mode however, the beam is focused into a fine probe, which is scanned across the specimen.

Once the desired electron beam is formed, it interacts with the specimen, which is placed in a dedicated specimen holder. This holder is located between the two pole pieces of the objective lens. The transmitted electrons are focused by the objective lens into a diffraction pattern in the back focal plane of the objective lens after which they recombine, yielding an enlarged image of the specimen in the objective lens its image plane. An ideal lens system is expected to image a single point source as a point.

Scherzer [1], however, demonstrated that for round symmetric electromagnetic lenses, aberrations are unavoidable. These aberrations contribute to blurring of the image and therefore a loss in resolution. For both the condenser- and objective lenses, stigmators are therefore present which apply an asymmetric, correcting field to minimize the effect of the aberrations.

Below the objective lens, a system of intermediate and projector lenses create a magnified image of either the sample in real space or the corresponding diffraction pattern in reciprocal space. This image can e.g. be visualized using a Charged Coupled Device (CCD) or a direct electron detector. The complete buildup of a transmission electron microscope is presented in **Figure 1**.

Version	Status	Date	Page
0	public	2023.03.20	10/34



**Figure 1:** Illustration of the components of an electron microscope [2]

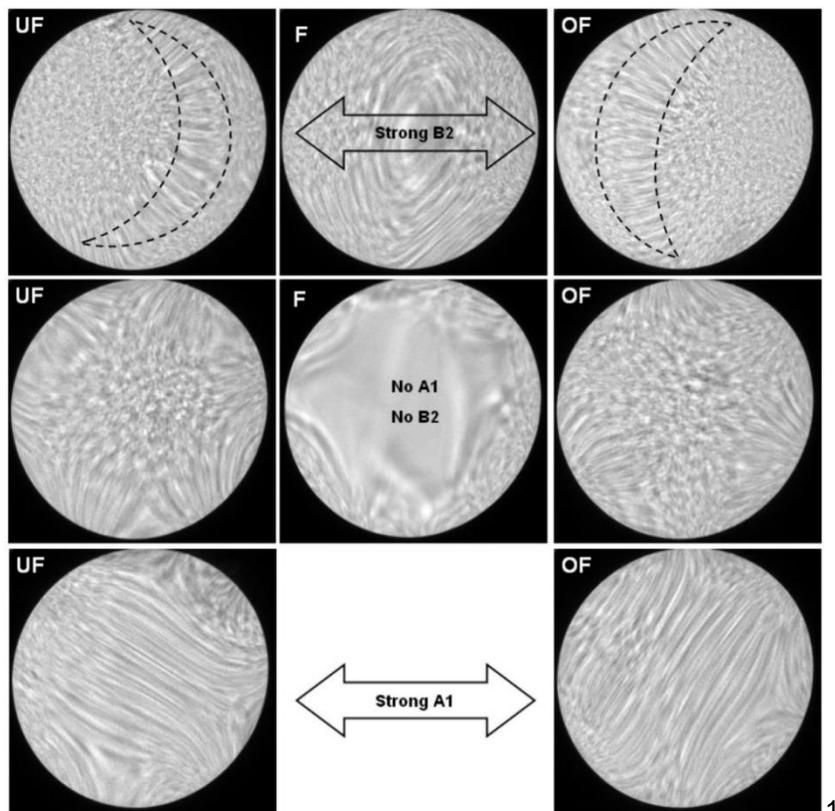
## 1.2 Ronchigram

For the STEM use case, Ronchigram images are key to finding the right parameters that minimize aberrations and therefore maximizes the spatial resolution. The term Ronchigram is a reference to the “Ronchi test”; a standardized test for shaping aberration-free optical lenses. By placing a diffraction grating within the focus of the optical lens, the imperfections of the lens would be recognizable from the obtained interference pattern.

Constructing a Ronchi’s grating is not feasible in TEM. Due to the high frequency of the accelerated electrons, the gratings’ spacing would have to be only a few picometers wide for interference to occur. Creating such a grating is therefore extremely challenging. Instead, the atomic arrangement in amorphous materials is used, which provide a nearly random assortment of atomic potentials. This random assortment forms a good approximation of a noisy grating which mimics the Ronchi test and gives interference patterns that reveal the presence aberrations in electromagnetic lenses.

Version	Status	Date	Page
0	public	2023.03.20	11/34

Key features in the Ronchigram which can be exploited to correct the lower order aberrations, consist of the Ronchigram’s symmetry and magnification. When in focus, the center of the Ronchigram has a high local magnification that represents the aberration-free portion of the electron beam. Further away from the center, thus moving away from the optical axis, aberrations reduce the local magnification. (**Figure 2**, center row). Having an as large as possible magnified central region is a first indicator for an optimal defocus. The presence of asymmetric aberrations breaks the rotational symmetry of the Ronchigram. Two-fold astigmatism unidirectionally stretches the region of high magnification, thereby producing distinctive streaks (**Figure 2**, bottom row). Axial coma, shifts the center of the Ronchigram and higher order aberrations further break the symmetry of the Ronchigram (**Figure 2**, top row).



**Figure 2:** Visualizations of a Ronchigram obtained on amorphous carbon in underfocus (UF) and overfocus (OF) as a function of astigmatism (A1) and axial coma (B2). [3]

If we want to train a reinforcement learning agent to correct for the lower order aberrations, it is essential that we have large amounts of labeled Ronchigram images. Obtaining such images would require substantial microscope time which is expensive. To levitate the need for physical system time we will therefore explore Digital Twinning (DT) as an alternative to create labeled Ronchigram data.

Version	Status	Date	Page
0	public	2023.03.20	12/34

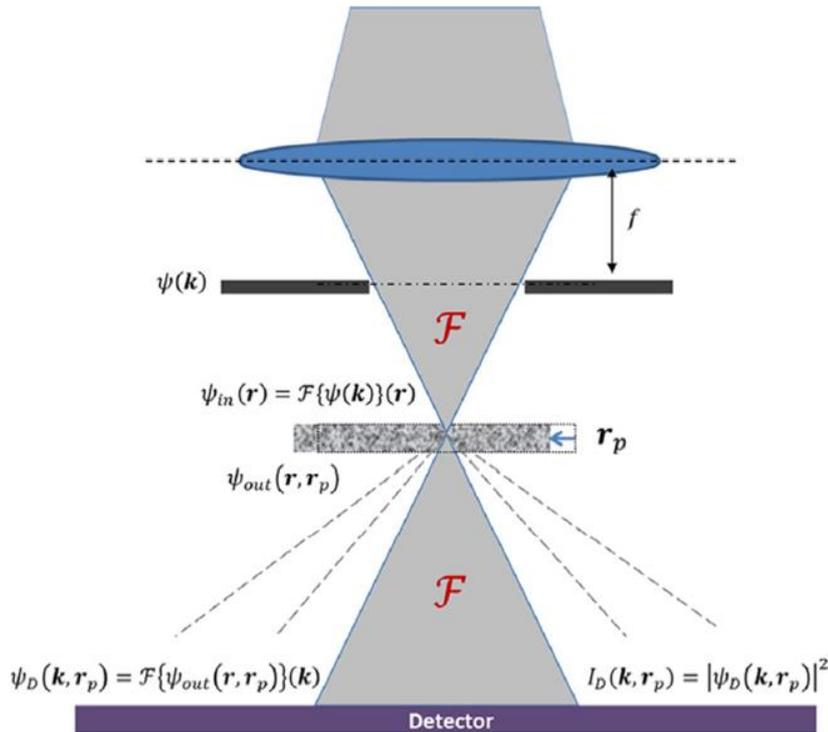
## 2. Digital Twin

A Digital Twin (DT) is a digital representation of a real-world system. The core of a DT consists of a model or a combination of models mimicking a generic real-world module or process. These models can be physics-based or data-driven and they are validated against the theory and the real-world data. Although we mostly work with the generic models at this stage, it is worth mentioning what a full-scope DT can look like. In its full scope, a DT is specific to a particular instance of its real-world counterpart and is predictive in a defined window into the future. In general, this requires the DT models or their parameters to be fine-tuned and for the DT to have real-time connection to the physical system. The latter requires an efficient DT. A DT with the above properties is capable of dynamically controlling the real-world system. More details on the properties and evolution of a digital twin can be found in [1].

A digital twin has several potential applications. Evaluating new ideas where the required hardware is not available is an instance where a DT can be used. Here, however, we focus on the fast and scalable data generation used in training a reinforcement learning (RL) algorithm. A TEM consists of several modules. The digital twin of TEM includes models describing these modules. Here, a wave-optics approach is used to describe the changes in the electron wave as it travels from top of the TEM column (the electron gun) until it interacts with a camera or a detector at the bottom of the column. For the STEM use-case, we target aberrations in the probe forming part of the TEM (i.e., the condenser system) wherefore a convergent beam electron diffraction pattern (i.e., CBED or Ronchigram image) is recorded on a pixelated camera.

TO DELETE [2]:

Version	Status	Date	Page
0	public	2023.03.20	13/34

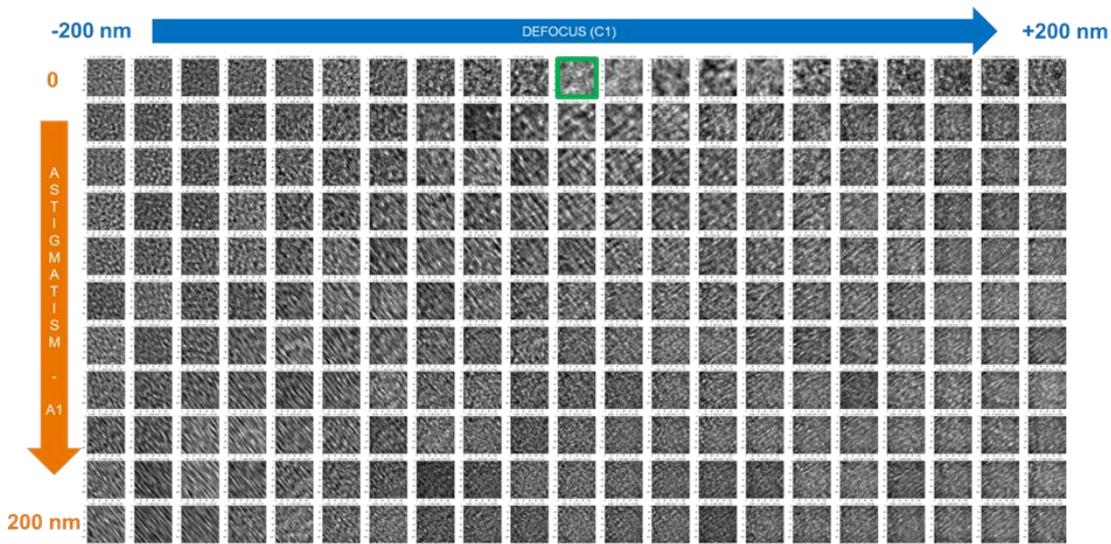


**Figure 4.** A simplified view of STEM imaging mode in a TEM and the CBED pattern formation. Figure from [3].

Using the DT described above, several datasets were generated, ranging from 80,000 to 800,000 images, which served as training data for the RL. For the objective of ASIMOV, the AI agent should learn how to move the state of a TEM system, via control knobs for first order aberrations, namely, defocus and two-fold astigmatism, towards a state with minimal aberration. Therefore, the data consists of Ronchigram images under different values of defocus and two-fold astigmatism.

A subset of typical DT data is shown in **Figure 6** with changes in defocus represented along the x-axis and changes in two-fold astigmatism magnitude along the y-axis. Here a central part from the Ronchigram has been cropped out to enhance the features of interest. The green box represents the state of minimal aberration, i.e., the goal for RL agent.

Version	Status	Date	Page
0	public	2023.03.20	14/34



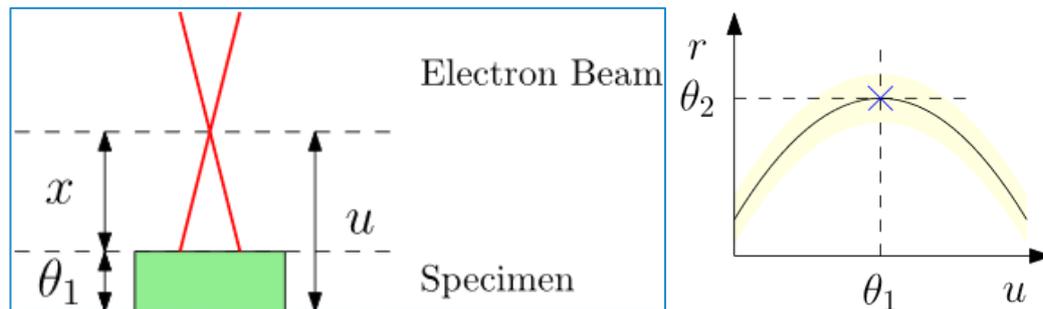
**Figure 6.** Changes in the Ronchigram due to defocus and two-fold astigmatism. The green box shows the goal state of zero aberration.

Knowing how synthetic data can be simulated, thereby alleviating the need for (time) expensive microscopy time, we will explore optimization strategies for retrieving the conditions with minimal lens aberrations and therefore the highest image resolution. Prior to discussing Reinforcement Learning strategies, we will investigate the performance of classical optimization problems on synthetic data.

Version	Status	Date	Page
0	public	2023.03.20	15/34

### 3. Classical optimization for aberration correction

The first step in applying classical methods to correct two-fold astigmatism and defocus is finding a valid abstraction. A very generic abstraction is the following; in calibration problems, the objective is to drive certain internal states  $x$  to zero. These internal states are ‘imperfections’ or aberrations in the case of the STEM. Once these imperfections are gone, the instrument is calibrated. An illustrative example of this abstraction for the electron microscopy use case is shown in **Figure 8**.



**Figure 8.** Abstraction of the focus problem (left) with focus state  $x$ , input  $u$ , and specimen thickness  $\theta_1$ . On the right, a curve constructed from observations of the focus problem, which resembles the reward  $r$  for different inputs. Observe that multiple inputs can result in the same value for  $r$  (i.e., the map is non-injective).

The figure on the left shows a simplified model of the focus problem. The electron beam is focused into a single point, which ideally should be on the surface of the sample to be examined. We can change the vertical position of the focus point through calibration, which is represented in the **Figure 8** by  $u$ . Additionally, there might be an unknown set of parameters  $\theta$ , which causes the calibration to be potentially different every time, hence the need for re-calibration. In this case, the internal state is the distance between the focus point and the sample surface,

$$x = u - \theta_1$$

Generally, this internal state cannot be directly measured (this would trivialize the calibration problem). Instead, we are only provided with an output or observation which gives indirect information about the internal state. For example,

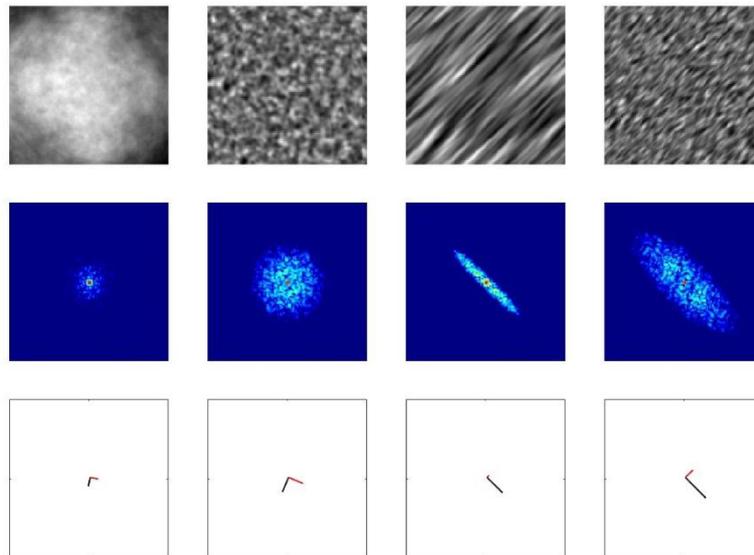
$$r = x^2 + \theta_2$$

where  $\theta_2$  is some unknown parameter.

Version	Status	Date	Page
0	public	2023.03.20	16/34

In this example, a single observation does not yield a unique corresponding internal state. In practice, an example of this in STEM is that an object in under- or overfocus is blurred in a similar manner. However, with multiple observations, it is possible to estimate the internal state or to move to an output which corresponds to the goal state. This exact type of problem clearly holds true for the STEM use case.

In this preliminary approach, only the defocus and the A1x tuning knobs will be calibrated. These two turning knobs correspond to two internal states: the x component of A1 astigmatism and defocus. The observation in the EM is the Ronchigram, which is a greyscale image. While it is possible to tune the two knobs in order to find a specific (category of) image, the problem can be simplified by instead evaluating a lower-dimensional feature space. As shown in **Figure 7**, the optimal input settings correspond to the Ronchigram which has the most axisymmetric and smallest geometry. These desired properties in the Ronchigram can be mapped to features through a series of operations. Therefore, the lower dimensional space of choice is constructed by applying a two-dimensional Fourier Transform to the Ronchigram, then fitting a two-dimensional Gaussian Multivariate to the resulting shape. The feature extraction process is shown in **Figure 9**.



**Figure 9.** Feature extraction process of the Ronchigram. The top row shows Ronchigrams corresponding from left to right to fully calibrated state; underfocus state; astigmatism with slight underfocus state; astigmatism with severe underfocus state. The second row shows the corresponding 2D Fourier transforms. The third row shows the eigen decomposition of the estimated bivariate Gaussian covariance.

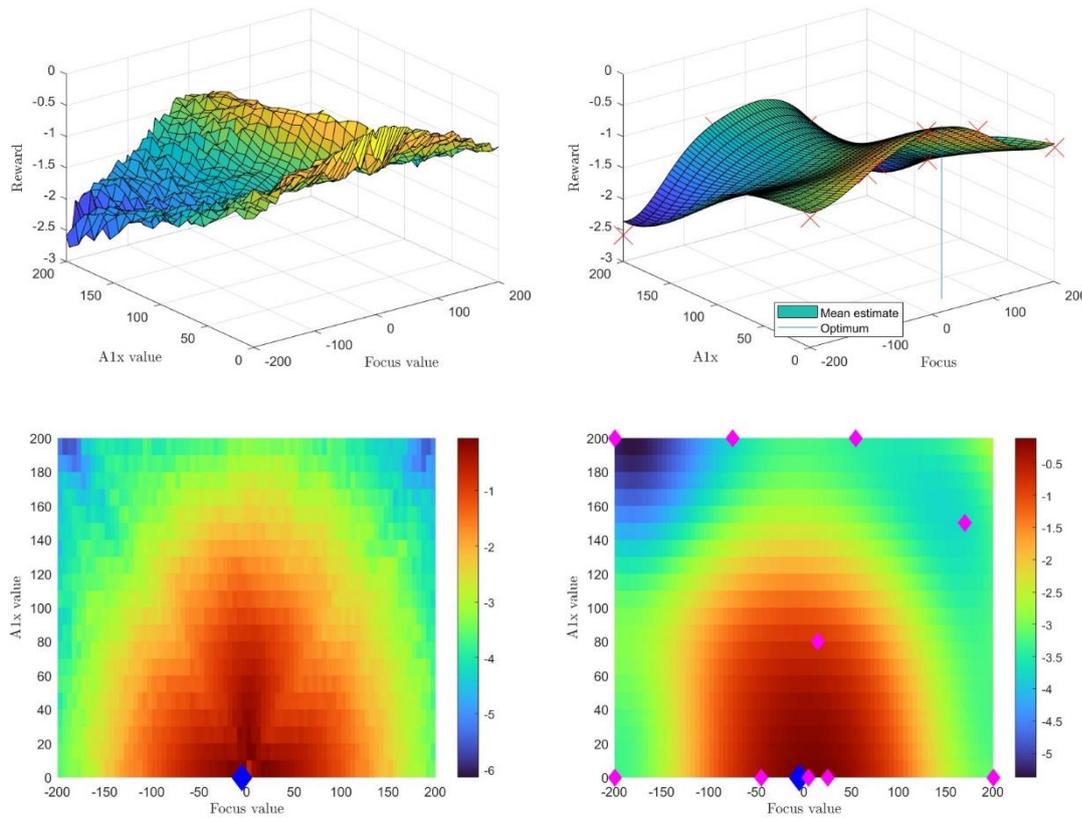
Version	Status	Date	Page
0	public	2023.03.20	17/34

From here, a scalar reward can be constructed from the eigenvalues of the bivariate Gaussian as a measure of quality of the current Ronchigram. In short, the calibrated Ronchigram corresponds to two small eigenvalues, while any deviation towards larger eigenvalues indicates a non-calibrated state. We can calculate the reward for a set of Ronchigrams according to a gridded set of inputs, which is shown in **Figure 10** on the left. Observe from the figure that the maximum reward (approximately) corresponds to the calibrated state  $x = 0$ , and that any distance from this state results in a lower reward.

If the complete set of Ronchigrams is available from the start, and a reward is calculated for each, finding the calibrating input is trivial. Instead, a more representative problem is to calibrate using a limited set of observations, which are chosen in a clever way. To do so, we simultaneously estimate the function we are maximizing. Gaussian processes (GP) present a method to estimate unknown functions effectively using very few samples. An application of GPs to estimate the reward environment is shown below on the right. This application uses a zero-mean prior and squared exponential prior covariance. The hyperparameters of the GP are optimized using log marginal-likelihood during the sampling of the function. After only 12 observations, the Gaussian process estimate largely resembles the rough shape of the true reward environment. More importantly, the maximum in the true reward environment coincides with the maximum in the estimate.

An additional benefit of this Gaussian process is that it also provides a measure of uncertainty for all points in the estimation. In automating the calibration problem, there is a trade-off between gathering information about the reward landscape (improving our estimates), and moving to where we currently believe the maximum is according to our GP estimate. One method to solve this trade-off is called Bayesian optimization (BO) [4]. BO balances between the estimated maximum and the uncertainty of the estimate in a so-called acquisition function. The results shown later in this section use the ‘expected improvement’ acquisition function, present in literature [5].

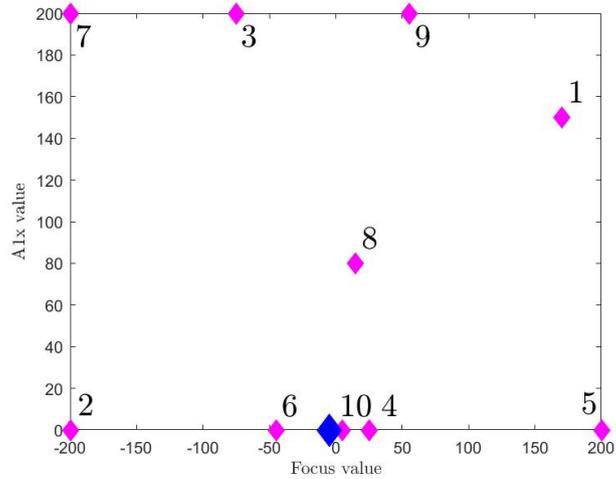
Version	Status	Date	Page
0	public	2023.03.20	18/34



**Figure 10:** Calculated reward landscape, based on the on dimensional feature extraction (left). Derived reward landscape by the Gaussian Process methodology (right). On top we visualize a 3D rendering from which one can appreciate the landscape, below we show the height profile of the landscape.

The results of applying BO with a GP estimator and the aforementioned feature extraction from the Ronchigram are quite promising for the use case. Using the synthetic dataset, a success rate (success = arriving at the calibrated state) of 100% in 50 attempts is achieved. In these attempts, an average number of 8.5 observations is required in order to arrive within a Euclidean distance of 10 from the calibrated state. In **Figure 11**, a typical episode is shown, where the marked points indicate the method's chosen inputs. The number indicates the corresponding input index. Note that the first input is randomized, and that the method is initialized without any datapoints at every attempt. Observe that Bayesian optimization uses the first few observations to explore the edges of the input space, since those points have the highest uncertainty at first. After these points, the basic shape of the reward environment is already established. The later observations are used to gather more information around the true optimum of the reward environment (the point 0,0), since the BO acquisition function deems this area most promising.

Version	Status	Date	Page
0	public	2023.03.20	19/34



**Figure 11:** Illustration of the explored settings for the tuning knobs during a typical attempt.

Whilst advanced classical optimization can clearly solve the defocus/two-fold astigmatism problem, it is dependent on feature extraction and devising a reward landscape. This might become problematic when attempting to correct higher-order aberrations. It furthermore remains unclear how scalable the current method remains under such conditions: the initial exploration of the high dimensional space might become too time intensive. For this reason, we will explore the use of Reinforcement Learning (RL) to solve the same issue, since RL has already conducted the exploration during training instead of during application. Before discussing the experimental outcome, we will provide a short introduction on RL and a simplified example to clarify the core concepts of the performed work.

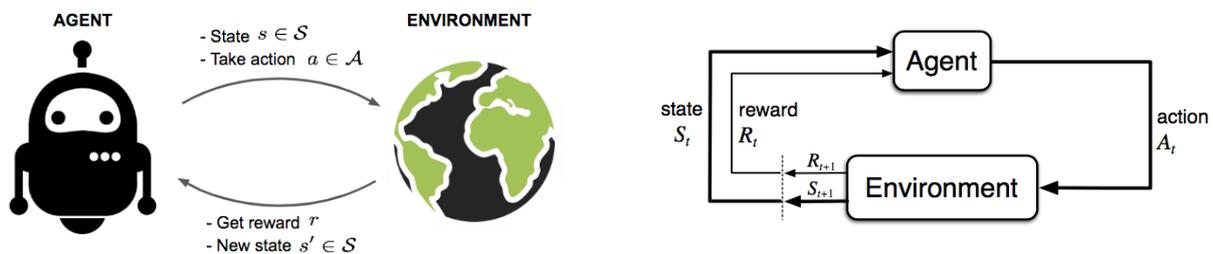
Version	Status	Date	Page
0	public	2023.03.20	20/34

## 4. Reinforcement Learning

### 4.1 Introduction

In short, RL refers to solving a particular type of problem where decision making is sequential, and the feedback is typically very sparse. Examples include computer games or robotics. A reinforcement agent interacts with a (virtual) environment and aims to attain both short- and long-term goals. For this, it is provided with a reward signal by the environment. What sets RL apart from traditional approaches such as from the optimal control field or optimization is its ability to learn from very sparse feedback and to operate without any prior or learned assumptions on the environment or the reward landscape.

ASIMOV report **Task 1.1** contains an elaborate introduction to both RL and its main disadvantages, next to a bottom-up introduction to model free Q learning. [9]



**Figure 12:** Illustration of Reinforcement learning way of working. [10]

We will briefly touch upon the major challenges when applying RL algorithms. These are:

- Exploration-exploitation dilemma,
- Data inefficiency,
- General performance of RL
- Reward function design,
- Stability and repeatability.

The exploration-exploitation dilemma refers to the fact that the RL agent must learn from its behavior (exploitation) applying what it knows, yet also explore new actions to take to expand its knowledge. (exploration) In this sense, this dilemma is similar to the exploration issue for Gaussian Processes (Section 3) and can be difficult to balance. In addition, RL agents are (hopelessly) data inefficient. A sparse reward signal, meaning that you only receive a reward once you have reached the goal, translates into massive data requirements, stressing the need for simulated data. Not only must this data be available, but it also results in long computationally intensive training runs. Designing more complex reward functions, as a non-trivial and potentially subjective task. Lastly, RL training is subject to high variance and general instability. Together with the aforementioned data inefficiency, it makes research tough.

Version	Status	Date	Page
0	public	2023.03.20	21/34

Prior to going into detail on the application of reinforcement learning on the simulated and experimental Ronchigram data, we will present the ASIMOV-in-a-Nutshell demonstrator. This demonstrator uses a simplified system that is inspired by the electron microscopy use case, in order to provide initial insights into the TEM use case.

#### 4.2 ASIMOV-in-a-Nutshell demonstrator

The goals of this demonstrator are as follows:

- Illustration of the ASIMOV project vision in a simpler setting than the TEM use case.
- Early exploration of the impact of Digital Twin modelling decisions on Reinforcement Learning.
- Early exploration of the systems engineering challenges imposed by Reinforcement Learning.

For fast exploration in terms of both development time and runtime without high hardware requirements, the demonstrator ignores detailed physics modeling and case-specific image recognition as much as possible. In the remainder of this section, we first introduce the simplified system to be optimized. Afterwards we explain the experimental set-up and the resulting agent behavior. Finally, we discuss some early exploration results beyond topics that are already addressed in the Electron Microscopy use case.

##### 4.2.1 System to be optimized

The ASIMOV-in-a-Nutshell demonstrator optimizes a noisifier system that produces images with two types of noise (blur and dots). Such abstraction was chosen to have a comparable system to the defocus and astigmatism deterioration of TEM images. The system produces noise images based on a collection of non-noisy base images, and two knobs that control the applied noise levels. At the beginning of each episode a base image is selected randomly, and afterwards the system can be operated by repeatedly observing the produced noisy image and by adjusting both knobs.



**Figure 13:** Illustration of the ASIMOV goal

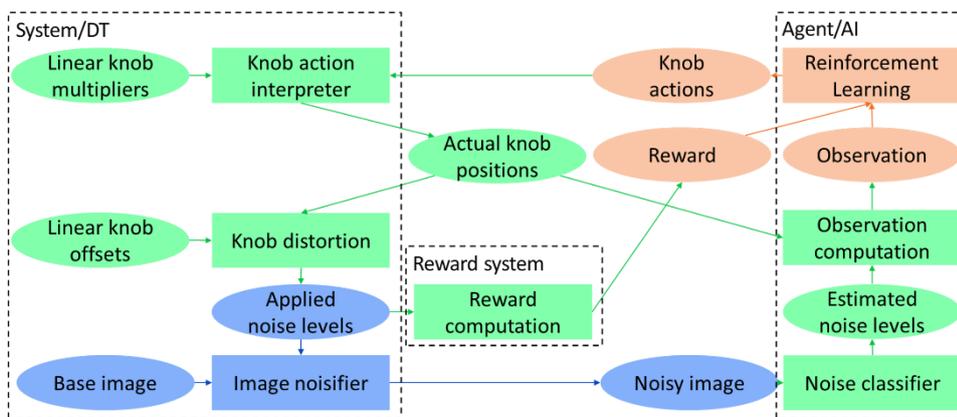
As shown in **Figure 13** the goal is to develop an RL agent that operates the system knobs until an image with minimal noise is obtained. To this end the optimizer can only use the input knobs and the output images. In particular, the optimizer cannot reduce the noise by applying some post-processing to the produced noisy images. Note the similarities with the electron microscopy use case. In both cases there is an imaging system with control parameters, and when the control parameters are not set optimally the

Version	Status	Date	Page
0	public	2023.03.20	22/34

result is a noisy image. Of course, the type of noise differs, both in the way it is computed and in the way it can be recognized in the images.

4.2.2 Experimental set-up

The experimental set-up for the ASIMOV-in-a-Nutshell demonstrator is summarized in **Figure 14**. We use rectangles to represent processing steps and ovals to represent data artifacts. The processing steps are grouped in three dashed areas conform the ASIMOV reference architecture [11]. Note that we use a single area for the physical system and digital twin, and that we ignore data orchestration such as logging.



**Figure 14:** Asimov-in-a-Nutshell Experimental Set-up

The blue elements are directly related to the image noisifier. Its output is a noisy image, and its inputs are the collection of base images and the applied noise levels. When all applied noise levels are set to 0, the noisy image is identical to the base image.

The red elements are directly related to the Reinforcement Learning technology that we intend to use for the optimizer agent. Its outputs are knob actions (moving the knobs in a certain direction or to a certain position), and its inputs are the observation and reward. The reward is only used during the training phase; the other inputs and outputs are used both in the training and operational phase. We use the Soft Actor-Critic (SAC) algorithm [7] for Reinforcement Learning.

Finally, the green elements represent the intermediate components that link the blue and red elements:

- The noise classifier (or regression) is a separate AI component that takes a noisy image and estimates the applied noise levels. It is based on a LeNet-5 convolutional neural network that is trained using supervised learning using only the image noisifier.

Version	Status	Date	Page
0	public	2023.03.20	23/34

- The reward computation is only used during the training phase, and it accesses the real applied noise level. This information would not be externally visible on a real implementation of the noisifier system, but it is accessible in a DT of the noisifier system.
- The knob distortion computes the applied noise levels based on the actual knob positions. To ensure that the agent does not know the optimal knob positions in advance, the knob distortion adds linear knob offsets that are selected randomly at the beginning of each episode.
- The knob action interpreter computes the actual knob positions based on knob actions, which can be either relative changes in a certain direction or absolute positions. To experiment with different knob sensitivities between multiple similar systems (for example within a product family, or between a real system and a digital twin), it also applies linear knob multipliers that are selected randomly at the beginning of each episode.
- Finally, the observation computation takes the estimated noise levels and actual knob positions as observation input of the reinforcement learning. To be more precise, it takes both the current values and the previous historical value.

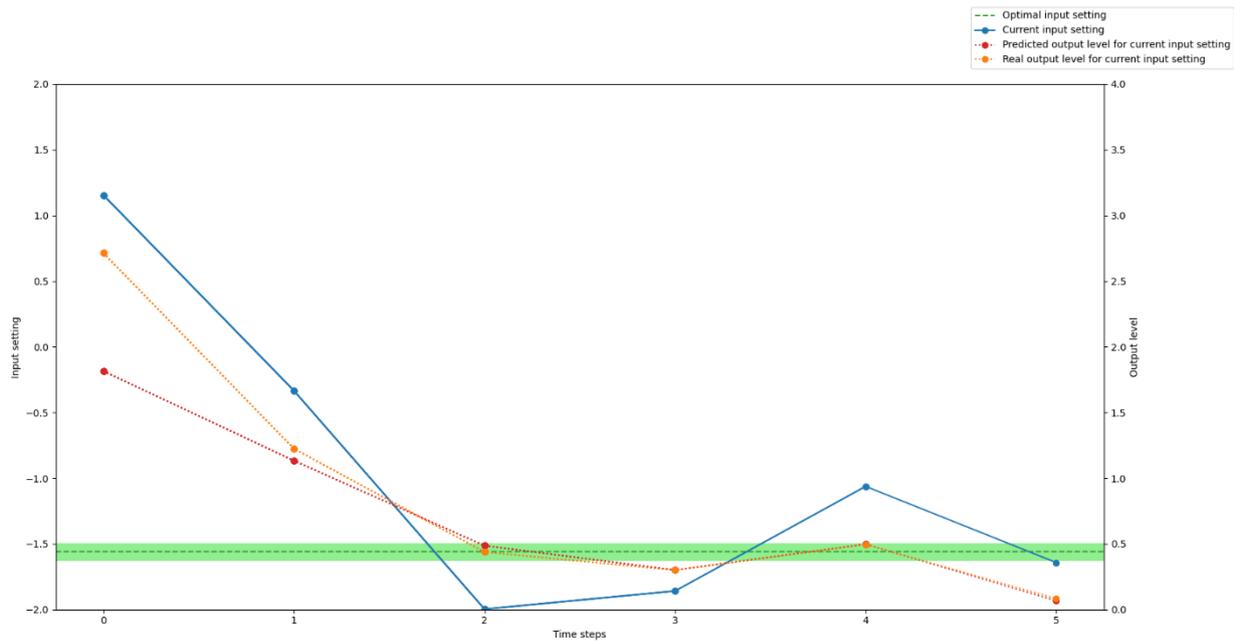
Our current implementation consists of two independent copies of the agent and reward system, where each copy focuses on one specific noise type.

#### 4.2.3 Agent behavior

When running the trained agent on this system, for each type of noise we obtain diagrams like **Figure 15**. The horizontal axis denotes time progression. In addition, there are two vertical axes:

- The left vertical axis denotes knob positions (ranging from  $-2$  to  $2$ ). The green dashed line indicates the optimal knob position, and the green area indicates the knob positions that are sufficiently close to the optimum. The blue line indicates the attempts by the agent to reach the green area. Note that the agent is not aware of the green area nor the green dashed line.
- The right vertical axis denotes noise levels (ranging from  $0$  to  $4$ ). The orange line indicates the noise level applied by the noisifier, whereas the red line indicates the noise level estimated by the agent. Note that the agent is not aware of the orange line.

Version	Status	Date	Page
0	public	2023.03.20	24/34



**Figure 15:** ASIMOV-in-a-Nutshell agent behavior

Experimentation shows that the optimizer always seems to reach the optimum area. The blue line from **Figure 15** shows the very typical swinging back-and-forth movements that stabilize around the green optimum. This mimics a typical human approach to find an optimal value.

Nevertheless, the number of action steps from the agent to get to the optimum area varies greatly. This raises questions about systems engineering for such AI agents. Do we understand the behavior of the agent? Do we need to understand the agent before we can trust it? How to deal with the varying number of required steps (including steps that make the system temporarily function less optimal)? Basically, such an AI agent is a relatively unpredictable component.

When comparing the difference between the orange and red lines, you can see that the noise classifier does not behave perfectly. Especially when there is either a lot of noise or very little noise. Moreover, when there are multiple types of noise, the accuracy of a specific noise classifier typically improves when the other noise levels get lower. The inaccuracies at very low noise levels cause challenges for the Reinforcement Learning agent when getting close to the green acceptable area.

To explore scaling the number of knobs, we have naively combined two independent agents for single knobs. We notice that the agents sometimes jump out of their green area after reaching it, as there is currently no clear stopping criterion for them. This shows that reaching the optimal area and staying in the optimal area are different things for the AI agent. It is further work to develop stopping criteria that

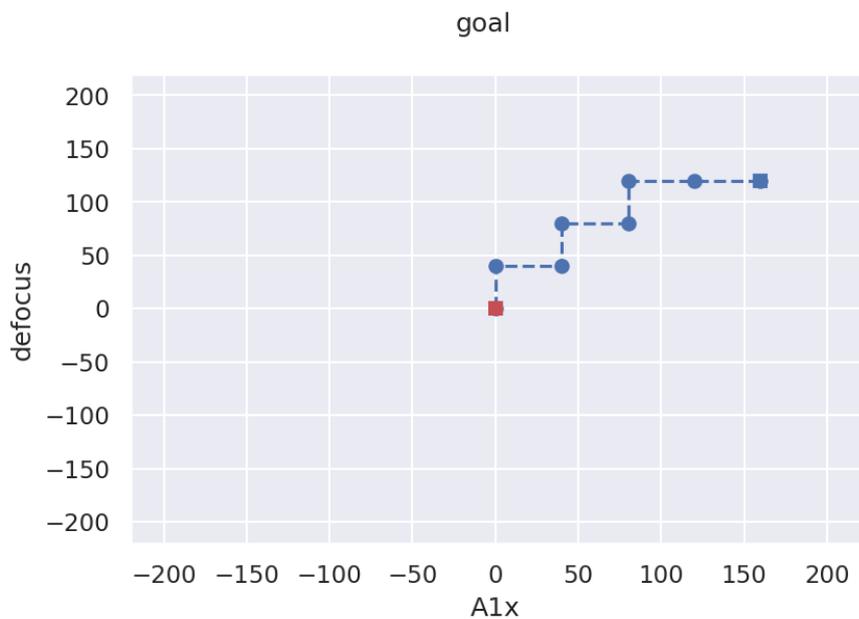
Version	Status	Date	Page
0	public	2023.03.20	25/34

give a more efficient workflow. Moreover, we envision that an RL agent trained on both knobs can reduce the number of steps to reach the desired goal and can display even more systematic paths.

Version	Status	Date	Page
0	public	2023.03.20	26/34

## 5. Reinforcement Learning for Aberration Correction

Having explored and clarified the initial concepts through the ASIMOV-in-a-nutshell demonstrator, we will go into more depth on the way-of-working for the electron microscopy use-case. Our deep learning approach was based on model free Q learning, on the simulated Ronchigram images. **Figure 18** shows an example roll out of a synthetically trained agent on unseen real data. Synthetic data performance is stable at 100%. The agent learns to handle the defocus knob to infer the A1x lens aberration. On average the Agent needs 12-16 steps to reach the goal. Real world data performance currently sits at 84%, but critically suffers from several issues that we will detail here. Firstly, the environment provides the goal state and the accompanying feedback to the agent. On an actual microscope the agent will have to learn how to stop by itself. Secondly, the environment defines the bounds of known universe for the agent and will simply put it back the moment it tries to venture outside. Although we don't fully observe this behavior, it is possible to get quite close to the goal state by learning to always turn the A1 knob up and then to bounce around.



**Figure 18:** Illustration of a trajectory taken to reach the optimum state on experimental data. The red square reflects the end state, whilst the blue square indicates the starting position.

Version	Status	Date	Page
0	public	2023.03.20	27/34

IR1.2

Proof of Concept Demonstration and Evaluation –  
use case electron microscopy  
public



Version	Status	Date	Page
0	public	2023.03.20	28/34

## 6. Live application at the electron microscope

Until now we have trained and validated the developed methodologies on simulated and/or pre-acquired data. In this section we discuss connecting the pre-trained RL agent to the TEM and enabling real-time application.

### 6.1 Discussion and Results

Using the described set-up, we connected the RL agent (**Figure 17**) to the electron microscope. In order to test the RL agent performance, we placed the microscope in various misaligned states by changing the value of defocus (C1) and A1x. For a pre-set number of steps, we acquired a Ronchigram, processed it through the Reinforcement [Learning](#) Network and applied the A1x and C1 changes suggested by the network.

Below we provide a graphical representation of the conducted experiments. The potential initial values for A1x were [0-, 50, 100, 150, 200, 300] nm while A1y was kept at 0. C1 was initially set to a value within [-300, -150, -50, 50, 150, 300] nm. It must be noted that the used RL network was trained on data for which the aberrations did not exceed 200 nm. In addition, it only encountered states for which the aberration was a multitude 20 nm. Hence, in our experimental set up, the network was pushed beyond its training conditions.

For each experiment, the network was allowed to conduct 15 steps. On the graphical representation (**Figures 20-21**) each experiment is illustrated as a line. The dot represents the end state of the experiment. The experiments have been color-coded in terms of their success. The measure of success was determined by the network's ability to get the microscope back to the aligned state (A1x = 0, C1 = 0).

To account for the fact that the best C1 value depends on the local thickness, the success condition was relaxed by introducing a tolerance. If the network drove the TEM to a state for which the Euclidian distance  $\sqrt{C_1^2 + A_{1x}^2}$  was below 40 nm (roughly 2 steps away from the 'goal'), it was counted as a success. If the end state of an experiment is within the tolerance, the experiment is a success and the line is colored green. If the experiment was at any point, apart from the start- and end-state, within the tolerance, it was considered a partial success and colored orange. If the experiment never came within the tolerance, it was deemed a failure and colored red. We must consider partial successes as well, since we currently do not have a stopping criterion. Hence, the network will keep performing steps even though it might have already encountered the 'goal' state. Hence, establishing a stopping criterion will turn the partial successes to real success.

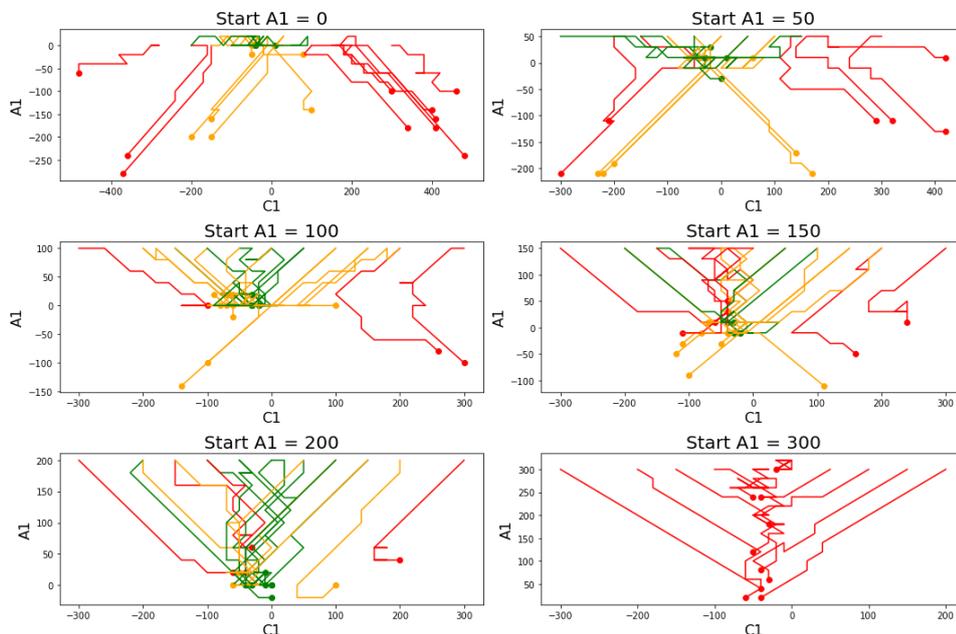
Version	Status	Date	Page
0	public	2023.03.20	29/34

In **Figure 20** we display the experimental runs as described above, separated as function of their initial A1x value. **Figure 21** illustrates a subset of the same runs as a function of initial C1 values.

A first thing to notice is that whenever an experiment was started with an A1x or C1 value beyond the trained conditions ( $\pm 200$ ), it has difficulty navigating towards the goal state. Either the network drives A1x or C1 in the wrong direction, or the network is unable to reach the desired state within the pre-set 15 steps. This demonstrates that the network does not necessarily generalize well over unseen conditions.

For every starting point with values below 200 nm, two experiments were conducted. It is curious to see that not in all cases they follow the same trajectory. For a starting value of A1x = 50 nm and C1 = 150 nm, we can even discriminate a case where one experiment succeeded, and the other failed. This is the result of the non-zero exploration factor at the end of training the Neural Network.

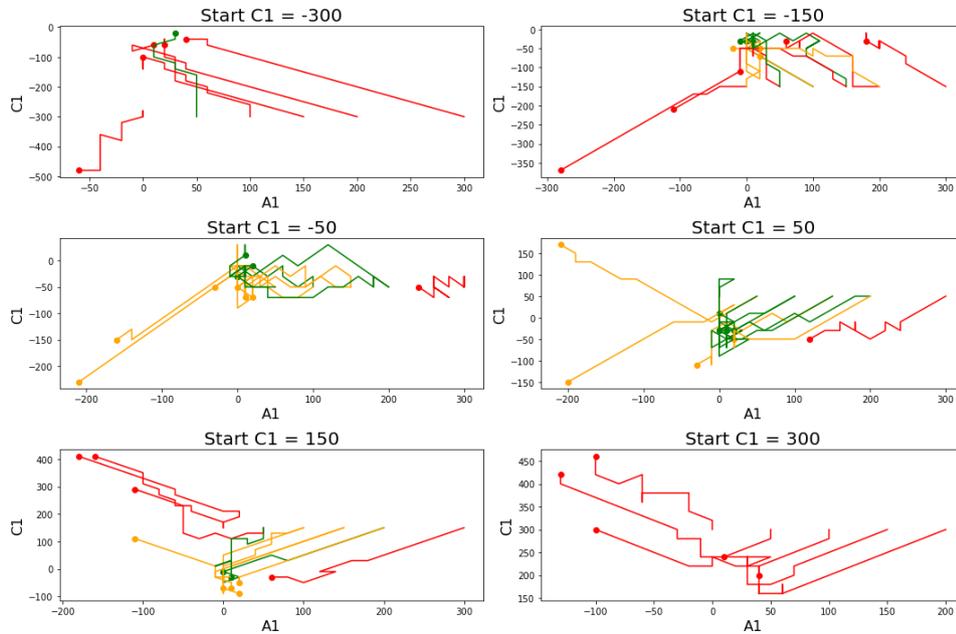
For high values of A1x ( $>50$ ), the network consistently knows in which directions it should drive C1 in order to reach the goal state. Experiments with a low initial A1x value but a high defocus, prove to be difficult as the network seems unable to establish whether it currently is in over or under focus. Overall, the high number of (partial) successes depicts a promising future for the TEM use-case.



**Figure 20:** Illustration of the RL trajectories as function of their starting A1x values. The trajectories have been colored as function of their performance. If the TEM is within the tolerance of the goal (0,0) for the last step, the trajectory is green. If the TEM is not within the tolerance for the last step, but came within

Version	Status	Date	Page
0	public	2023.03.20	30/34

the tolerance during operation, it is orange. If the TEM never came within the tolerance, the trajectory is red.



**Figure 21:** Illustration of the RL trajectories as function of their starting C1 values. The trajectories have been colored as function of their performance. If the TEM is within the tolerance of the goal (0,0) for the last step, the trajectory is green. If the TEM is not within the tolerance for the last step, but came within the tolerance during operation, it is orange. If the TEM never came within the tolerance, the trajectory is red.

### 6.2 Conclusion

Here we have demonstrated the potential of RL to align the electron microscope and reduce A1x and defocus. Although the pretrained agent generalized well on offline experimental data (85%), we could easily find starting conditions for which the agent would consistently fail. Hence, more effort is required to establish a robust and reliable workflow. First of all, a stopping criterion needs to be established to ensure that the network no longer acts once it has reached the optimum state. This can either be trained in the network or done in a less explicit way. In addition, until now only a small subset of aberrations has been corrected. In order to get a functional workflow, the RL agent needs to be scaled up to include more, higher order, aberrations.

Version	Status	Date	Page
0	public	2023.03.20	31/34

TO DELETE[8].

Version	Status	Date	Page
0	public	2023.03.20	32/34

## 7. Terms, Abbreviations and Definitions

*Table 1 - Terms, Abbreviations and Definitions*

ABBREVIATION	EXPLANATION
AI	Artificial Intelligence
DT	Digital Twin
EM	Electron Microscopy
RL	Reinforcement Learning
STEM	Scanning Transmission Electron Microscopy
TEM	Transmission Electron Microscopy
C1	Defocus
A1	Two-fold Astigmatism
A1x	X component of two-fold astigmatism

Version	Status	Date	Page
0	public	2023.03.20	33/34

## 8. Bibliography

- [1] O. Scherzer, *Zeitschrift für Phys*, vol. 593, p. 101, 1936.
- [2] “<https://open.oregonstate.edu/general/microbiology/chapter/microscopes/>,” [Online]. [Accessed 23 11 2022].
- [3] FEI, “FEI Application Instructions: How to tune STEM probe on Cs-corrected Titan.”.
- [4] M. G. Kapteyn, J. V. Preterius and K. E. Willcox, “A probabilistic graphical model foundation for enabling predictive digital twins at scale,” *Nature Computational Science*, vol. 1, pp. 337-347, 2021.
- [5] E. J. Kirkland, “Advanced computing in electron microscopy,” *Springer Nature*, 2010.
- [6] I. Lazić and E. G. T. Bosch, “Analytical review of direct STEM imaging techniques for thin samples,” *Advances in Imaging and Electron Physics*, vol. 199, pp. 75-184, 2017.
- [7] D. Jones, M. Schonlau and W. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, vol. 13, no. 4, p. 455–492, 1998.
- [8] E. Brochu, M. Hoffman and N. de Freitas, “Portfolio Allocation for Bayesian Optimization,” in *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.
- [9] H. N. a. H. La, “Review of Deep Reinforcement Learning for Robot Manipulation,” in *Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, 2019.
- [10] “<https://lilianweng.github.io/posts/2018-02-19-rl-overview/>,” [Online]. [Accessed 11 23 2022].
- [11] ASIMOV-consortium, “ASIMOV Reference Architecture,” 2022.
- [12] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, 2018.
- [13] T. McDermott, D. DeLaurentis, P. Beling, M. Blackburn and M. Bone, “AI4SE and SE4AI: A Research Roadmap,” *INCOSE Insight*, vol. 23, no. 1, pp. 8-14, March 2020.

Version	Status	Date	Page
0	public	2023.03.20	34/34