

ASIMOV Cookbook

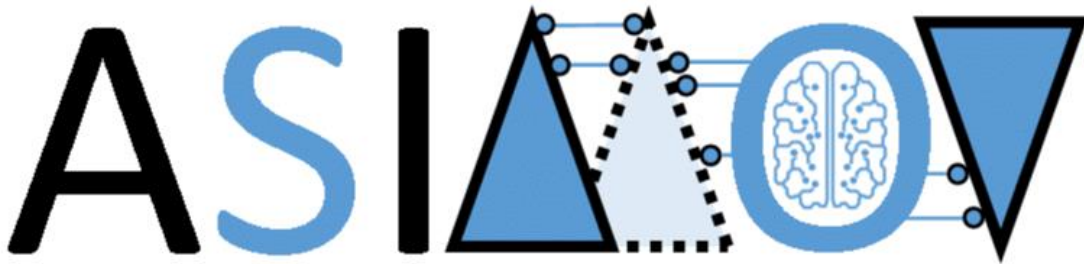
Rewards: School of AI and Digital Twins

What am I getting into?

June 1st, 2024

Editors: Bram van der Sanden, Richard Doornbos, Jan van Doremalen

Contributors (alphabetically): AVL Deutschland GmbH, Consultants in Quantitative Methods CQM B.V., Eindhoven University of Technology, Thermo Fisher Scientific, German Aerospace Center (DLR), LiangDao GmbH, NorCom Information Technology GmbH & Co. KGaA, RA Consulting GmbH, The Netherlands Organisation for Applied Scientific Research (TNO) - Embedded Systems Innovation (ESI), Triangraphics GmbH



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



ASIMOV is an ITEA project in the Eureka cluster on software innovation, AI call of 2021.

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

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 PARTY, IN WHOLE OR IN PARTS, EXCEPT WITH THE PRIOR WRITTEN CONSENT OF THE ASIMOV CONSORTIUM. THIS RESTRICTION LEGEND SHALL NOT BE ALTERED OR OBLITERATED ON OR FROM THIS DOCUMENT. THIS PROJECT HAS RECEIVED FUNDING FROM THE ITEA4 JOINT UNDERTAKING UNDER GRANT AGREEMENT NO 20216. THIS JOINT UNDERTAKING RECEIVES SUPPORT FROM THE EUROPEAN UNION'S EUREKA AI RESEARCH AND INNOVATION PROGRAMME AND FINLAND (DECISION PENDING), GERMANY, THE NETHERLANDS.

Abstract

In this ASIMOV cookbook we have gathered best practices and lessons learned identified by the entire consortium. This document provides a starting point for strategists, R&D managers, system architects and system engineers that are faced with the question: “How to build complex high-tech cyber-physical systems that select their optimal settings autonomously within minimal time and with minimal external expertise?”. In the cookbook, we will give guidance on how to assess whether there is a viable business case, and how to get started. We also give an overview of our experience in constructing an ASIMOV solution, focusing on the overall architecture, as well as the AI and digital twinning parts. We also provide a framework to help with the question on how to embed the solution in the organization, to turn the initial solution into a mature and successful result.

Contents

1	Introduction	4
1.1	Cookbook introduction	4
1.2	Use cases	5
1.3	Digital twinning	6
1.4	Artificial Intelligence and Reinforcement Learning.....	6
2	Business Case	8
2.1	Current situation: what are the problem characteristics?.....	8
2.2	Business ambitions and value proposition.....	8
2.3	Will AI and digital twinning bring value to your problem?	9
3	Getting started	11
3.1	Identify the path to value	11
3.2	Ensure that the organization is ready	11
3.3	Ensure that domain knowledge is available	11
3.4	Ensure that the right data is available	11
3.5	Start simple	11
3.6	Evaluate early	12
3.7	Setup the development process	12
4	Digital Twinning	13
4.1	Accuracy and type.....	13
4.2	Data processing	13
4.3	Operating modes.....	14
4.4	Validation	14
4.5	Dealing with drift.....	14
5	Artificial Intelligence	15
5.1	Reinforcement Learning methods.....	15
5.2	Best practices in establishing a training strategy.....	15
5.3	Lessons learned and best practices	15

6	ASIMOV architecture: AI + Digital Twinning	17
6.1	<i>Best practices to set up the solution architecture</i>	17
6.2	<i>Bridging the “reality gap”</i>	17
6.3	<i>How to make trade-offs?</i>	17
6.4	<i>How the solution properties are determined by digital twinning and AI aspects</i>	18
6.5	<i>Infrastructure</i>	19
7	Embedding in the organization (2 page)	20
7.1	<i>AI Maturity Framework</i>	20
7.2	<i>Strategy</i>	21
7.2.1	<i>Business</i>	21
7.2.2	<i>Business case</i>	21
7.2.3	<i>Tactics for introducing AI and DT</i>	21
7.3	<i>Data and Technology</i>	22
7.3.1	<i>Architecture</i>	22
7.3.2	<i>Performance and qualities</i>	23
7.4	<i>People and Organization</i>	23
7.4.1	<i>Organizational aspects</i>	23
7.4.2	<i>Starting development</i>	24
7.5	<i>Governance</i>	24
7.5.1	<i>Regulations and legal aspects</i>	25
8	Conclusion	26
9	Terms, Abbreviations and Definitions	27
10	Bibliography	28

1 Introduction

1.1 Cookbook introduction

Focus in this cookbook: autonomous and self-optimizing CPSs High-tech cyber-physical systems (CPSs) play increasingly important roles in our society. They are ubiquitous, and companies, organizations and societies depend on their correct functioning. CPSs need to have high up-times, be user-friendly, and economical to use. CPS suppliers must assure that their systems reliably deliver optimal quality in customers' environments, without bothering their customers with complex system optimization tasks that require highly skilled staff. Systems need to be optimally tuned before delivery and at installation and re-adjusted during use, which can easily require many hours/days and this total time increases rapidly due to growing project diversity and complexity. To address this major business issue, it is ASIMOV's vision that CPSs must be increasingly autonomous and self-optimizing.

Within the ASIMOV context, we focus on high-tech CPS that have the following typical characteristics:

- they have limited availability, and are expensive in use;
- executing actions is time-consuming, and can also potentially be dangerous;
- they are typically complex, making it hard for a human to fully understand their behavior and emergent properties;
- they need a highly skilled technician to be (re)calibrated.

Why this cookbook? With this cookbook, we provide guidance to strategists, R&D managers, system architects and system engineers to decide whether a solution with AI and digital twinning might be a solution with clear benefits to an optimization problem in high-tech CPS. The cookbook also provides a starting point for system architects and system engineers to answer the following question:

"How to build complex high-tech CPSs that select their optimal settings autonomously within minimal time and with minimal external expertise?"

What's in the cookbook? This cookbook describes lessons learned, gained insights, and best practices developed within the ASIMOV research project (2020-2024). The focus is to report on innovative technologies, combining AI and digital twinning, that can be used to create autonomous and self-optimizing CPSs. In such systems, AI-based software autonomously performs system optimization tasks during development, integration, manufacturing, installation and use by the customer. The approach is shown in Figure 1.1 and includes the following key innovations:

- How to create digital twins of systems to simulate realistic system behavior;
- How to train an optimization-AI based on the digital twin to find optimal system settings;
- How to verify the validity of the digital twin to train the AI;
- How to use the trained AI to perform the tuning and calibration tasks on actual system configurations.

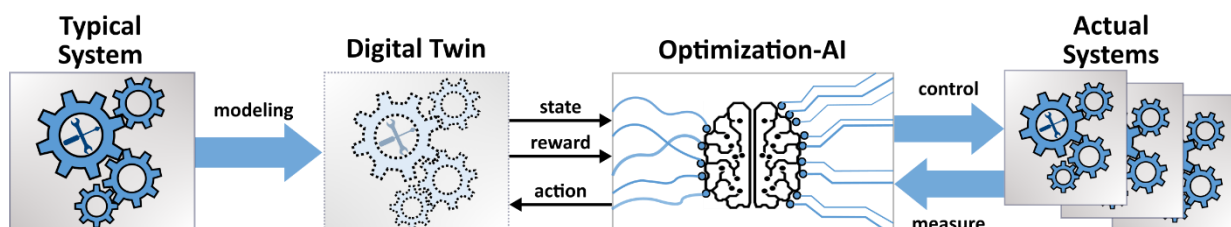


Figure 1.1 The ASIMOV solution - Digital-Twin based training of AI to optimize system settings.

Use cases The lessons learned and best practices are based on the following use cases taken from two different industrial system domains for which optimization is crucial for system performance:

1. Automatic lens calibration in a Transmission Electron Microscope (TEM);
2. Automatic test case generation to test Unmanned Utility Vehicles (UUVs).

1.2 Use cases

Automatic lens calibration in a TEM (TEM use case)

The first use case focuses on calibrating the lenses in a TEM. Transmission electron microscopes use electron beams instead of light to magnify and image ultrathin specimens in the nanometer range. Important application domains are biology, for example studying bacteria and viruses, chemistry, for example to research nano plastics, and semiconductor research. A transmission electron microscope is shown in Figure 1.2¹. In electron microscopes, the electron beams are affected by magnetic fields, and the behavior of electrons is influenced by their wave nature. To achieve high-resolution imaging, electron microscopes employ a series of electromagnetic lenses to focus and manipulate the electron beam effectively. In scanning mode, the electron beam is focused into a fine probe, which is scanned across the specimen. To get clear images, the electromagnetic lenses should be calibrated. Calibrating a microscope requires a lot of training, and it is very time consuming, taking weeks.

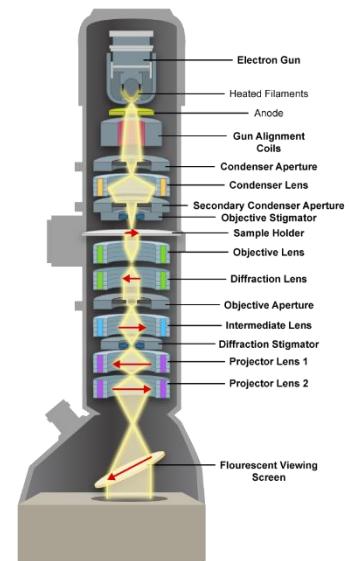


Figure 1.2 Illustration of a TEM

In ASIMOV [1], an AI solution is used to automatically calibrate an electron microscope. Training such an AI on real microscopes is too expensive, as these systems are very expensive, not readily available as they are very heavily used, and they have unique configurations. Instead, digital twinning is used to train the AI. The digital twin mimics how a test specimen (a Ronchigram) looks like under different lens configurations. Using this digital model, part of the lens calibration can be done without needing time on the actual system.

Automatic test case generation to test UUVs (UUV use case)

The second use case focuses on autonomous vehicles driving without a human driver. An illustration is shown in Figure 1.3². Examples are small shuttle busses that provide on demand public transport or UUVs that deliver goods. As these vehicles are driving on the public road, safety is a key aspect. There are two important challenges in ensuring safety: (1) There are many possible road situations that the vehicle should be able to deal with. Next to that, these kind of autonomous vehicles are typically tailored to a specific purpose. As consequence, the vehicle safety needs to be validated for a variety of individual vehicle configurations.



Figure 1.3 Illustration of an UUV

The ASIMOV solution is used to address these two challenges [2]. Digital twinning and AI are used to setup a well-calibrated and comprehensive test bed. Furthermore, digital twinning is used to mimic a wide range of road situations, calibrated with real-world scenarios. Through the use of DT environments, also highly dynamic and potentially dangerous situations can be simulated, without any safety concerns. The UUV is driving in this virtual world, and tests performed in this setup give insight into the vehicle driving characteristics, including whether the UUV is driving as intended and reacting safely to unexpected situations. An AI is trained to select which test scenarios should be run to give most insight in testing and validating vehicle safety, reducing overall testing time. This custom tailored testing strategy pinpoints the individual weaknesses of a vehicles' perception and driving function setup.

¹ Figure source: <https://open.oregonstate.edu/generalmicrobiology/chapter/microscopes/>

² Figure source: <https://www.continental.com/de/presse/pressemitteilungen/2019-12-12-hmi-cube/>

1.3 Digital twinning

In our ASIMOV solution, we refer to the system that we would like to optimize as the **physical twin (PT)**. A **digital replica** mimics the PT. We distinguish between a digital model, digital shadow, and digital twin according to the definitions suggested by Aheleroff [3].

- **Digital model:** a digital replica of a system (the PT) which is (more or less) static, for example a simulation or mathematical model. It mimics the behavior of the system without the requirement to change as the system changes over time.
- **Digital shadow:** a digital replica that has a real-time link from the PT. If the state in the physical twin changes, then this state change is reflected in the digital replica in real-time.
- **Digital twin (DT):** a digital replica that has a bi-directional real-time data connection with the PT.

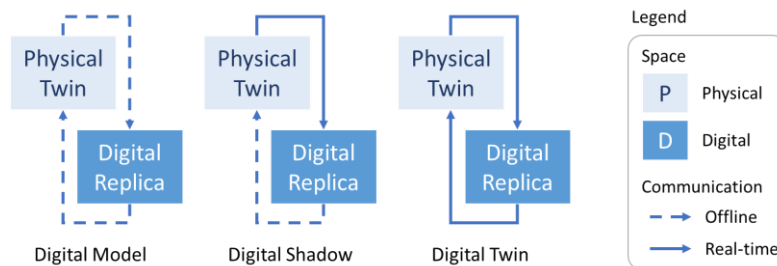


Figure 1.4 Digital twinning integration levels (adapted from Fig. 7 in [3])

1.4 Artificial Intelligence (AI) and Reinforcement Learning (RL)

Artificial intelligence. AI has evolved tremendously since the early 2010s. In recent years, data has proven to be a key enabler to enhance many engineering systems through AI.

In the ASIMOV solution, the focus is on RL, where an RL agent is trained using digital twinning and used on the Physical Twin (PT). This choice for RL was made for three key reasons:

1. traditional approaches to control and optimization problems had already been extensively considered in the ASIMOV use cases and these had already been stretched to their limits;
2. the nature of the optimization problem is sequential decision making [8], where the final goal is long-term, and the aim is to get to that goal as fast as possible; and
3. other machine learning solutions require supervisor interaction, which would be highly impractical in the industrial use cases considered, as these systems are supposed to operate autonomously or with minimal supervisor interaction.

The research with RL was started therefore with the idea that significant breakthroughs might be reached.

Reinforcement Learning. RL is not a mature field; there are many scattered methods and variants, and researchers refer to the same methods by different names. In our work we followed more closely [4] rather than the standard reference [5], although methods from [5] were also highlighted. This choice was motivated by the fact that [4] considers mostly simulation-based settings whereas [5] focusses on experimental-based setting. Therefore [4] is closer in spirit to the present project ASIMOV. However, many important methods are not covered in [4] and for those we rely mostly on the survey paper [6].

Reinforcement learning (RL) solves a particular range of problems, where decisions need to be made sequentially and the final goal depends on what happens in the long term over many steps. RL aims to learn good policies from experimental trials and feedback from the environment. The policy takes as input the current state and gives as output a decision. Ultimately, the learned policy should lead to decisions that maximize cumulative rewards in the environment. What sets RL apart from other algorithm families such as optimal control and many optimization methods is its ability to learn a policy with no modelling beyond the description of states, actions/decisions, and using the environments rewards (costs and benefits).

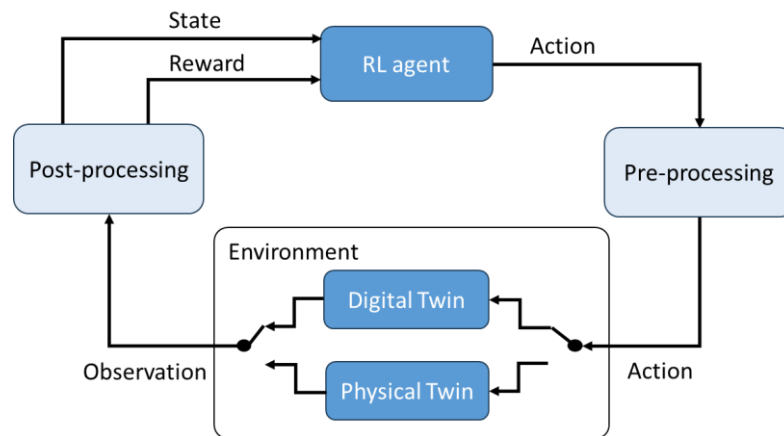


Figure 1.5 RL agent interacting with the system (adapted from Fig. 1 in [7])

Figure 1.5 shows the key concepts for RL in relation with the environment:

- An RL agent acts in an **environment**. This environment can be either the **digital twin** or the **physical twin**.
- The agent takes sequential **actions** based on the **state** of the environment it observes, after **post-processing** of the **observation**, and the decision policy, which action to take in which state, it follows.
- The environment changes based on the **pre-processed** action that is taken by the RL agent. The environment delivers a new state and gives feedback on costs and benefits realized, the **reward**. Usually, there is randomness involved in which a new state materializes, influenced by the action.
- The principal **goal** of the agent is to take decisions that maximize the sum of discounted future rewards. The reward will be higher if a more desirable state is achieved and so the agent learns to take actions that move the system towards its longer term optimal goal.
- Each action in the environment leads to a new state. In each state, the environment delivers a reward. In general, learning updates are only done at the end of an episode. An episode is a sequence of state and reward observations and actions in line with a policy. The information contained in the episode is used to adjust the policy.
- There might be differences in the structure of the observations, depending on whether the environment is the digital twin or physical twin, leading to differences in the post-processing. Similarly, there might also be differences in the pre-processing.

2 Business Case

2.1 Current situation: what are the problem characteristics?

To assess whether an ASIMOV solution might be applicable, a clear view is needed on the system, what is the (optimization) problem at hand that you want to solve, what are the problem characteristics, and why is it currently difficult to tackle the problem. It is also key to identify the usage scenarios where the problem manifests itself, and which ones have the biggest (negative) impact. Identifying a clear and concise problem statement helps to develop a targeted and effective solution, and it fosters a common understanding among the stakeholders.

With the ASIMOV solution, we focus on the problem of system optimization in high-tech CPS. This problem often has the characteristic that there is a clear tuning goal, but the tuning steps to get to that goal are not clear upfront. As such, the problem nature is **sequential decision making** [8], where the final goal is long-term, and the aim is to get to that goal as fast as possible. Typical challenges are the amount of domain knowledge required, the time to get to an optimal solution, and the high operational cost of the system.

Identifying the usage scenarios related to the problem is essential to zoom into specific instances where the optimization problem manifests itself, and to determine when the optimization is performed.

2.2 Business ambitions and value proposition

Given the current situation and the problem statement, it is important to determine where your business ambitions are and what is the value proposition. What are the system properties you want to improve, and what is the envisioned business value that it will bring? These system properties can then be used to identify the most important usage scenarios to address. Having these use cases helps to develop a targeted solution. For each use case, the relevant stakeholders should be identified and what their expectations are from the solution.

With the ASIMOV solution, we target the following functional and non-functional properties:

Property	Definition
<i>Functional properties</i>	
Accuracy of result	Does the PT state after applying the optimization achieve the required accuracy.
Robustness	Ability of a system to operate effectively and reliably under various conditions, including disturbances (external disturbances coming from outside of the system) and extreme inputs.
Reliability	Ability to always obtain a good result.
Reproducibility	Ability to always obtain the same result.
Time to result	Execution time of the solution in operational phase to reach the result.
Scalability	Ability to be usable in product family, and for a variety of usage scenarios.
Explainability	Ability to give insight into how the solution works, and plausibility of the result.
Safety	The PT should never exhibit unsafe behavior.
<i>Non-functional properties</i>	
Footprint	The amount of space or energy that the solution needs.
Integrability	Ease with which the solution can be embedded into the existing product.
Maintainability	Ease with which a system can be maintained, repaired, or updated over time in the field.
Development cost	The total expenditure associated with designing, creating, and implementing the ASIMOV solution

With the ASIMOV solution, we focus on systems that are currently optimized by a human operator and where in the envisioned situation the system is automatically optimized using reinforcement learning (RL) and digital twinning.

Next to the system properties, it is useful to consider what level of autonomy is envisioned with an AI solution that autonomously optimizes the system. As framework, the levels of automation from the

ISO/IEC 22989 standard [9] can be used, where level 6 is autonomous, and the other levels are heteronomous, with human involvement.

Level of automation	Description
6 Autonomy	The system can modify its intended domain of use or goals without external intervention, control or oversight.
5 Full automation	The system can perform its entire mission without external intervention
4 High automation	The system performs parts of its mission without external intervention
3 Conditional automation	Sustained and specific performance by a system, with an external agent being ready to take over when necessary
2 Partial automation	Some sub-functions of the system are fully automated while the system remains under the control of an external agent
1 Assistance	The system assists an operator
0 No automation	The operator fully controls the system

2.3 Will AI and digital twinning bring value to your problem?

The following lists help to decide whether a solution with RL and/or digital twinning could bring value to your problem situation.

When to go for an ASIMOV solution?

- **When system optimization needs to happen often.** If system optimization is only done occasionally, then no automated solution is needed.

When to go for AI?

- **When automation is important**
- **When the solution needs to deal with many variations**
- **When the solution needs to deal with drift**
- **When it is not possible or infeasible (due to complexity) to use a classical algorithm**

After deciding whether to go for an AI-based solution, the next question is to decide upon the techniques that could be used given the problem's nature. For example, for classification and regressions problems, there are many supervised learning approaches available. For clustering problems, unsupervised learning techniques can be used. For (sequential) decision making, RL is a typical approach. The following list gives some guidance on when to go for RL, and when not.

When to go for RL?

- **When the problem's nature is sequential decision making;** i.e., optimization requires making multiple dependent consecutive decisions. RL is suited for problems where decision making is sequential, and the final goal is typically long term.
- **When highly trained experts are needed to optimize the system.** RL automates system optimization, decreasing the need for scarce, highly trained experts to optimize the system.

When not to go for RL?

- **When there is an easier solution to address the problem.** For example, if a system model can be made, and the problem can be tackled using dynamic programming or any other optimization technique. Another example is if the next state follows from the previous state and action in a deterministic, non-random way.
- **When a system model and data are available, and the aim is to solve a global optimization problem.** Then, methods to use include Bayesian optimization, Tabu search, simulated annealing, and TSP algorithms. These techniques are well-known, and there is a lot of knowledge available on how they work, including usage scenarios illustrating their benefits and drawbacks for specific situations.
- **When you don't have the real system or a digital twin to interact with.** Training an RL agent requires a system to interact with. In some cases, instead of a digital twin, a prepared data set can be used. For example, in the TEM use case, we could use a data set with images for each system state to train the RL agent, assuming a limited number of states. In general, however, this is often not possible.

- **When the problem nature is making a single isolated decision**, not affecting subsequent decisions. In that case, supervised machine learning can be used.

Predicting up front whether RL will be successful is difficult due to various reasons. The technology around RL has not yet matured, for example the software available to train RL agents. There are also not many successful use cases where RL has been applied on a physical system with an RL agent inside.

RL can also be used with other AI techniques. For example, in the famous Alpha Go example, RL was combined with heuristic search techniques. When working with images, one can for example use AI-based image recognition techniques and feed those results to the RL agent. AI techniques can be used for hyperparameter optimization of RL, for example Bayesian optimization.

When to go for digital twinning?

- **When the real system is not always available.**
Digital twinning decreases the time to optimize, as the RL-agent can be trained using the model instead of the real system that might not always be available.
- **When the real system has high operating costs.**
AI is data hungry. It is cheaper to use a digital twin for training than a complex CPS that typically has high operating costs.
- **When there are potential safety issues in using the system.**
Using a digital replica is safe compared to using a physical twin, and as such it provides clear advantages to use for dangerous tasks and early exploration.
- **When training time is important.**
A digital twin typically simulates the behavior of the real system faster than the real system executes, decreasing the time to train an RL agent.
- **When there are many system variations.**
A digital replica is easy to configure, and can provide large amounts of diverse, labeled data. Having data for different variations also enables the RL to build upon this experience.

When not to go for digital twinning?

- **When the real system is readily available, and inexpensive to use.**
- **When training time is not an issue.**
- **When developing a digital replica is infeasible or more expensive compared to using the real system.**

In these situations, the high cost to develop and maintain an accurate (enough) digital replica can be avoided, and the real data from the system can directly be used to train the RL agent.

3 Getting started

3.1 Identify the path to value

To deliver tangible and measurable value to the stakeholders, it is important to start with the end goal in mind and identify the path to value. It includes setting key milestones and ensuring that each step contributes to achieving the end goal. We have experienced that an agile way of working works very well. It is important to quickly identify and learn from failures (failing fast) in the early stages, as it helps to identify the key challenges and mitigate risks by addressing potential issues early on.

3.2 Ensure that the organization is ready

Developing a solution including digital twinning and AI requires specific competences. The organization should be ready, having data engineers, AI specialists, and architects with a basic understanding of AI and digital twinning technologies. Embedding the solution into the organization is discussed in more detail in Section 7.

3.3 Ensure that domain knowledge is available

It is important to have domain knowledge available. Especially in the early phases, domain knowledge is needed to interpret the results provided by the RL and digital twinning, and to validate them. Domain knowledge is also essential to determine what is the right data. Furthermore, domain knowledge is needed to develop a digital replica that accurately mimics the behavior of the system.

3.4 Ensure that the right data is available

Having the right data available is key to calibrate and validate the digital twin (see also [10]) and subsequently also to train the RL agent. It can be hard to define up front whether the right data is available. It helps to start by examining what data is available, for example what is the *type* of data available, for example the structure, form, and shape, and whether *meta-data* has been logged; e.g., what are the system settings, mode, use case related to the data at hand? Next, the available data can be assessed using the following criteria (see [11] for more details):

1. **Representativeness:** does the data cover the right diversity of parameters, frequencies, and errors? As a rule of thumb one can try to assess whether all information required by a human expert to solve the problem is present.
2. **Volume:** what is the data volume? Is there sufficient data to train a data-driven solution like RL on?
3. **Bias:** are some aspects over- or underrepresented in the data?
4. **Completeness:** are all aspects of the problem well-represented in the data?
5. **Quality:** are there systematic errors introduced by the data collection process? For example, are there signs of data duplication and data corruption.

Once there is an initial data set, it is important to validate whether the data is sufficient (e.g., the *right data* is available), and for which situations additional data is needed.

- Does the digital replica mimic the physical system accurately enough after calibration using the data? Can the digital replica generate similar data?
- Does the RL trained using the digital replica behave as expected? In which situations does the RL agent make undesired decisions?
- Can additional required data be generated on demand in an online setting, when interacting with the PT? In other words, can the PT be used in dedicated experiments?

3.5 Start simple

An important lesson learned is to start small and go for a ***minimum viable solution***. Decide on a restricted use case with a clear target audience and with a clear set of stakeholders. In this way, feature creep can be avoided, and it is much easier to show the potential value quickly.

Nowadays, there are many frameworks available that support the development of an AI and digital twinning solution. For example, machine learning platforms to build and deploy AI models, and containerization platforms to run a digital replica efficiently in the cloud. Using such frameworks means that you do not have to develop all these functionalities yourself. However, note that it is important to consider the (lack of) maturity of technology and suppliers. There is a risk of lock-in, especially once the solution starts growing over time, and a higher associated migration cost to other platforms.

3.6 Evaluate early

The solution with digital twinning and AI should be evaluated early. We therefore recommend an iterative and agile development approach. This will guarantee quick feedback on whether the quality of the digital replica outputs to the RL agent is sufficient to optimize the system, in which cases it performs well, and for which cases more/better data or improved data processing is needed. Thus, we move from partial to complete coverage of the problem scope and from feasibility to optimality. It also will allow for a quick assessment of whether the approach will lead to an effective and efficient solution to the problem.

3.7 Setup the development process

We have identified four steps in developing an ASIMOV solution [11], shown in Figure 3.1:

1. **Initial development:** an initial digital model of the typical system is created, for example a first-principle model, using data and domain knowledge. Also, the AI infrastructure is set up for training and fine tuning in subsequent steps.
2. **Connect digital replica and AI:** the digital replica is connected to the optimization AI, in our setting the RL agent. The RL agent is trained using the digital replica.
3. **Connect AI and CPS:** the digital replica is replaced by the real CPS to allow the trained RL agent to exercise control over the CPS.
4. **Connect AI and other CPSs:** other CPSs of the same product family can be connected to the trained RL agent to enable the trained RL agent to exercise control.

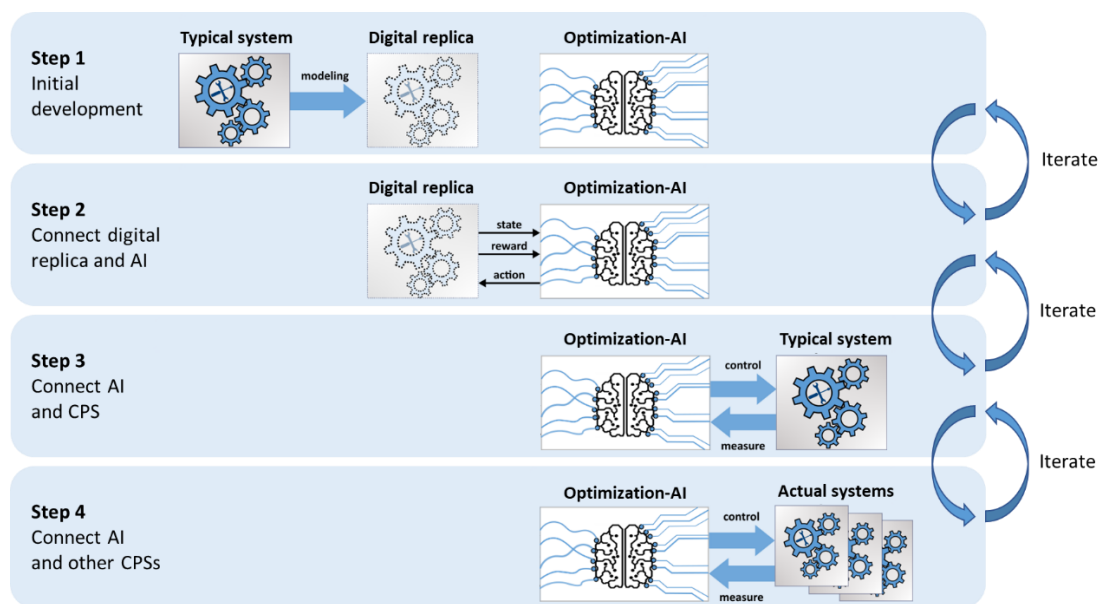


Figure 3.1 Development steps of an ASIMOV solution

When setting up the development process, it is important to ensure that you have the infrastructure ready from the start, such that it can grow simultaneously with developing the solution.

4 Digital Twinning

Before starting to develop a digital replica, it is important to consider the lessons learned described in the following subsections.

4.1 Accuracy and type

Make the digital replica only as accurate as necessary. Depending on the purpose, the model fidelity doesn't necessarily need to be very high. Start with a simple digital model, as it likely needs to be tailored once the RL agent is trained on it. A simple model is easier to adapt and to understand to get good results in an early stage.

Switching fidelity in the digital model can be done along at least two important axes:

1. the level of detail considered in the digital model, and
2. the time step considered in the digital model [12].

Adding more detail to the model will increase the amount of computation power and storage needed. Changing the time step can for example be done inside a component, but this change should be aligned with the time step chosen across components. Changing the time step is typically easy, but it might lead to numerical instabilities where the solution does not converge any more if the time step chosen is too coarsely grained [13].

A digital model might be sufficient, depending on the purpose. What is needed starts with the goal. A digital twin or digital shadow is required if the agent needs to address a problem that is dependent on the exact configuration of the physical twin. If not, a digital model might be sufficient. A digital twin is needed

1. if the physical twin and the digital replica need to interact in real-time. For example, a vehicle in a testbed that goes uphill needs a real-time connection with the model to accurately reflect the increased resistance in the dynamometers that simulate the load the wheels experience due to inertia and friction;
2. when direct control of the physical twin is required; or
3. when safety of future system behavior needs to be predicted at runtime [14].

4.2 Data processing

Postprocessing needs to be done with domain knowledge in mind. When processing the data, one has to take care that the relevant information is not thrown away. Based on domain knowledge, one of the available options is to manually identify the features to extract from the data.

Postprocessing can be considered as a limit on the required accuracy. Data is post-processed, as the original raw data often contains too much detail to train the RL agent. If the output of the postprocessing does not change for a different output of the digital replica, then a higher accuracy might not be needed. We identified this in both ASIMOV use-cases, as explained in [7], Sec 6.2. In the TEM use case, for some scenarios, the amplitude of the Fourier transform (FT) of the original image proved to result in a better generalization of the RL to the real data, compared to using the raw data. In the UUV use case, the movement of all traffic participants in the simulated scenario was post processed to specific KPIs that quantify the criticality of the simulated scenario, compressing the data substantially.

Consider the interplay between data processing and dealing with variations. In data post-processing (shown in the pipeline in Figure 4.1), handcrafted features can be used to process the data. This step depends on the type of data, but not on the concrete data values. The neural net inside an RL agent automatically learning features, and in contrast, these features are dependent on the data values. As a result, if the share of learning features from the data in the neural net is larger than the post-processing done using manual features, the system is more affected by variations during training. More details are available in [7].



Figure 4.1 Data processing pipeline from raw data to an action.

Consider whether Markov property is present and preserved. When applying post-processing to compress data, one needs to be careful not to lose the Markov property by removing essential state information. If this happens, the RL agent's perception of the environment is not solely determined by the current state and action, violating the Markov property. More details are available in [7].

If the Markov property is not there in the first place, then there are techniques that help to get the Markov property. For example, one can add domain knowledge, use frame stacking to add context information, or use a replay buffer to estimate the right system state (for example using long short-term memory as neural network type).

4.3 Operating modes

Consider the different operating modes of a digital replica. Initially, the digital model is not connected to the physical twin. This means that faster than real-time training is possible, and that multiple digital replica instances (mimicking for example different instances of a family of systems) can be running in parallel. In this phase, a lower fidelity model might be sufficient, whereas later a higher fidelity model is needed. This requires that the digital twin components capturing aspects at multiple fidelity levels can be changed easily. It also means that the goals and metrics can differ depending on the system development phase. In later steps, the digital replica may be connected to a physical twin via real-time communication.

4.4 Validation

In general, it is difficult to give guarantees whether the digital replica will accurately resemble a (slightly) different physical twin configuration. However, we used the guideline that **the digital replica will likely work if the specific configuration falls within the boundaries of the parameter settings in the training data set or can be interpolated from the parameter settings.**

The purpose of the digital replica is to provide useful training data to train the RL agent. The “**end to end validation**” yielding one number how well the digital replica did for its purpose **can only be done once the RL agent has been trained** using the digital replica and applied to the physical twin. If equivalent subsystems of the digital replica and the PT exist, verification and validation at this sublevel can help strengthen the trust in the validity of the digital replica.

4.5 Dealing with drift

The physical twin could drift over time. **Be aware of whether drift is also considered and possibly accounted for in the digital replica.** Once the digital replica is connected to the physical twin, drift should be monitored, and if drift is detected, the DT should be calibrated using observations from the physical twin combined with active sensing. Active sensing means that you take decisions that use additional information on where you are in the state space, to ensure that you can synchronize the DT and PT.

5 Artificial Intelligence

5.1 Reinforcement Learning methods

An RL agent interacts with its environment by observing states and executing actions based on its policy. The reward signal reflects immediate performance, and the objective is to discover an optimal policy that maximizes the value function. This function represents the cumulative long-term reward as a function of the states and possibly of the actions.

Optimal versus approximate methods Classical RL relies on precise representations of value functions and policies, making it suitable only for small, discrete problems. However, many real-world problems involve variables with a vast or infinite set of possible values, such as continuous variables. In such cases, there is a need to approximate value functions and policies to address the challenges posed by the complexity of these problems. RL is therefore divided into several subfields [6]. First, there are **optimal and approximate methods**. For simple (low-dimensional) problems, tabular methods, i.e., having a precise representation of the value functions, can lead to optimal behaviour. However, since for high-dimensional problems, such as the ones addressed in ASIMOV, some form of approximation must be carried out due to the curse of dimensionality, we focus on approximate methods.

Types of approximations. Second, there are different kinds of approximate methods (as also described in [10]). Roughly, we can categorize them as **approximations in value space**, **approximations in control/policy space**, and **approximations in both value space and control/policy space** (such as actor-critic methods). The approximation technique can be based on (deep) neural networks [11] or other forms of parametric approximation (an architecture that is off-line fitted based on data), or on-line simulation based.

5.2 Best practices in establishing a training strategy

Generic best practices for establishing a good practical training strategy can be found in many publications. In [15] important steps are listed that we adapted for the ASIMOV context:

- **Establish a budget for developing a digital replica.** Determine the domain knowledge and data that is needed to develop a digital replica. Regarding the data, consider the type and amount of data to be used to calibrate the digital replica, and whether labeled data is needed. This is important for justification of the investments in the business case.
- **Appropriate data.** Select the right data to develop the digital replica and label it for the specific purpose. Carefully select the data that is relevant to the system optimization problem, based on domain knowledge.
- **Ensure data quality.** Accuracy and consistency of labeled, timeliness, and correctness of (real-time) behavioral data.
- **Be aware of and mitigate data biases.** These biases come from blind spots or unconscious preferences in the project team or training data. Diversity in the team or assessment by independent external experts may counter this problem.
- **When necessary, implement data security safeguards.** Security and confidentiality may pose important restrictions on your system setup and training. Be aware of government regulations and company policies.
- **Select appropriate technology.** The tooling, both hardware and software for capturing and managing data should fit the requirements on volume, speed and scale and flexibility of annotation.

These generic best practices also hold for the ASIMOV cases of digital twin-based AI training.

5.3 Lessons learned and best practices

When starting with an AI solution, the following best practices and lessons learned can be used as guidance.

- The RL agent is used to automate a system optimization done by a user. As a first step, the RL agent can **use the same control knobs as the user** uses. This is a good place to start, as already established interfaces can be used. It can also help with understandability and

explainability of the initial solution. After building up understanding, more control knobs can be considered.

- Training an RL agent or machine learning solution is typically compute-intensive. **RL agents and machine learning solutions can be faster than classical methods.** Namely, the set of possible decisions is small, the best decision can be chosen quickly. This is in contrast to a setting where each decision is evaluated using compute-intensive partial differential equations solvers or complex simulations.
- **It can be very beneficial to start small.** In the project, different approaches were used, where one approach created the AI solution in an agile way, and the other approach worked more in a waterfall approach. In the latter case, the integration phase proved to be very challenging. In the agile approach, it was easier to integrate the trained AI solution, have a solution early in the process, and incrementally extend the scope.
- **Consider explainability when choosing an AI solution.** RL mimics how a person learns to operate a system. For some RL algorithms like DQN, the Q-values and action maps can be used to explain the policy. This is a benefit compared to approaches like neural networks that make decisions based on thresholds computed using statistics, that are not always easy to interpret or explain.
- **Think about the stopping criterion up front.** RL does not know by itself when a solution is “good enough”. It helps to have domain experts involved at the start to define the criteria for a stopping criterion. Sensor data or metadata can be taken into account during the system optimization process to determine whether the system is in the desired state.
- **Value-based versus gradient-based** We found that for deterministic problems, value-based algorithms typically work better, and for stochastic problems one could start with a gradient-based approach.
- There are many libraries for **hyperparameter tuning**. Hyperparameter tuning should only be done after deciding on more important design choices, like the type of neural network to be used and the data inputs. If the hyperparameter tuning is done by hand, the direct effect on the trainings can be better understood and more knowledge is built up.
- In the TEM use case, an RL agent trained for a particular type of system also worked well on other similar systems within the same product family. We did not yet experiment whether generalizing beyond the product family also works.

We also encountered some stumbling blocks:

- **For some cases the reward function must look ahead many steps.** This is for example typical when solving logistics problems, where the reward is only obtained once the full scheduling solution is obtained and has been used over a longer period.
- **Hyperparameters have a strong impact on the results achieved.** Hyperparameters are parameters whose values are used to control the learning process. Learning from literature on how to do this is cumbersome, as it is often difficult to reproduce results of papers. We experienced that it could take a lot of effort to tune the hyperparameters in the right way to get the same results.
- RL is typically used when the transition probabilities of the Markov Decision Process are not explicit. Often there is no “Markov property” guaranteeing that the future state of the system depends only upon the current state, and not upon earlier states. **Small breaches of the Markov property can dramatically undermine the performance of RL algorithms.** In the automatic lens calibration use case, multiple measurements are taken while changing the defocus parameter. Details about this “focus-stacking” approach can be found in [16].
- We learned that reward shaping is very complicated due to the learning nature of the RL and that it requires a lot of experimentation. Even though we followed all literature recommendations, we could not reproduce them all, especially the ones related to intermediate rewards.

6 ASIMOV architecture: AI + Digital Twinning

6.1 Best practices to set up the solution architecture

In setting up the architecture for an DT+AI solution, we identified the following best practices:

- **Containerization**, having containers as basic components, and configuration at one place. This provides *portability* of the solution to other platforms, and *scalability* by having the possibility to start multiple instances with low effort. Containerization also improves *maintenance*, as the container encapsulates both the application and all its dependencies. This helps avoid the “it works on my machine” problem that often arises due to version incompatibilities. Having images of containers also improves *reproducibility*, as the environment can be reproduced on another machine according to the same build steps in the image. Finally, it also improves **collaboration across teams/companies**, as it requires defining clear interfaces between containers.
- **Micro-service architecture**, where each micro service is *testable* in isolation and exchangeable for another service by having well-defined interface contracts.
- The **high-level reference architecture** described in [11] can be used to structure the solution architecture. As the starting point, a container can be created for each component.
- **Task queues and a worker architecture**, to increase *processing speed* using multiple instances (parallelism). Note that this comes at the cost of increased communication across the network. The state of the outstanding tasks is monitored through the queueing system.
- **Having multiple ways to interface with the system**, to improve *maintainability* including user interfaces to interact with the system, logging and status reports for status and result monitoring, and interfaces for configuration.

6.2 Bridging the “reality gap”

The discrepancy between the simulated environment using the digital replica and the PT is known as the reality gap. This gap can result from differences in, for example sensor noise or actuator dynamics, and can lead to declined performance when transferring a model from simulation to reality. This concept is in literature known as simulation to reality, for short “Sim2Real” [17], that focusses on transferring skills, knowledge, or models learned in a simulated environment to the real-world. There are various approaches to the problem of the reality gap:

- **Adapting the DT**: identify the gap between the digital replica and the PT, and see if the digital replica can be adapted to closer reflect the behavior of the PT.
- **Domain adaptation**: randomizing the domain in the digital replica, for example varying textures, lighting, object shapes, can help to generalize RL agent better to the real-world by exposing it to a wide range of variations during the training phase.
- **Transfer learning**: pre-training models using the digital replica, and fine-tuning them on real-world data from the PT can help to reduce the reality gap, leveraging the knowledge gained in the simulated environment with the DT.
- **Hybrid approaches**: combining data from the digital replica and the PT during training can help the RL agent to be more robust in real-world scenarios.

6.3 How to make trade-offs?

When developing a solution including digital twinning and RL, there are important tensions between digital replica and RL aspects. To resolve these tensions, trade-offs need to be made that (significantly) affect the (strongly emergent) resulting system properties. It is important to notice that trade-offs depend on the use case but could also depend on the phase. For example, the digital twinning development effort plays an important role during initial development, whereas the development effort in later stages is limited. Figure 6.1 shows typical key tensions (red dashed arrows) and aspects supporting each other (shown with green solid arrows).

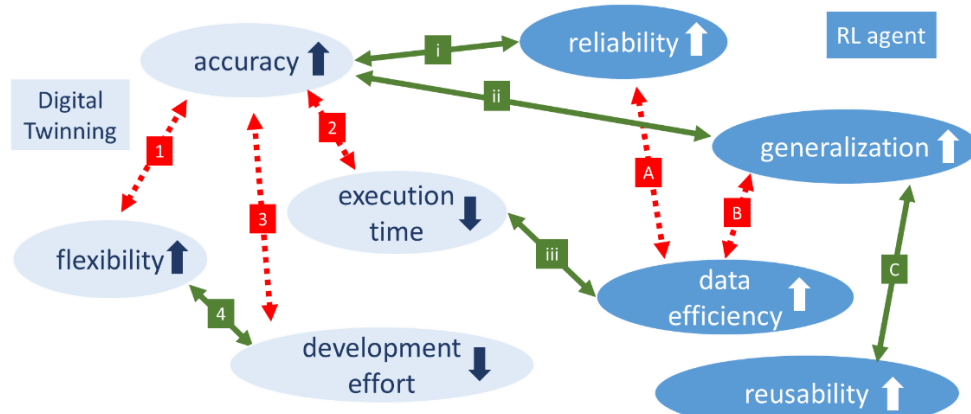


Figure 6.1 Key tensions between digital twinning and RL agent aspects.

In digital twinning, the goal is to have a high-enough accuracy or model fidelity to mimic the physical twin. A higher accuracy could mean that the digital replica is specific for an instance, reducing flexibility (tension 1). A higher fidelity also typically means an increased execution time (tension 2), and it requires more effort to develop the digital replica (tension 3). High flexibility of the digital replica is key to support different system variants, allowing for re-use, and avoiding the development of a completely new digital replica (support 4).

For an RL agent, a high reliability comes at the cost of requiring sufficient representative data to train on (tension A). A high generalization means good performance beyond the training environment. This improves reusability (support C), as the RL agent requires limited to additional training. It comes however at the cost of requiring more data to cover the various scenarios to train on (tension B).

A high digital replica accuracy supports high reliability of the RL agent (support i), as the RL agent is trained on an accurate digital model of the real system. An accurate digital replica, mimicking the real system in all relevant scenarios, also supports generalizability (support ii). It is important to consider the relevant scenarios up front, as the accuracy of the digital replica can only be determined in context of a real system and the relevant use cases. If the training process for the RL agent is very data efficient, then this reduced the execution time of the digital replica (support iii).

6.4 How the solution properties are determined by digital twinning and AI aspects

As part of the business case, we identified the key properties of the ASIMOV solution. The digital replica and AI components together have a big impact on these properties. The following table shows the relationship between digital twinning, AI, and the ASIMOV solution properties. Details on the relation to digital twinning aspects can be found in [10] and details on the relation to AI aspects in [16].

Property	Relation to digital twinning aspects	Relation to AI aspects
<i>Functional properties</i>		
Accuracy of result	The fidelity of the digital replica plays an important role in whether AI can bridge the DT-PS reality gap and is sufficiently accurate.	The stopping criterion is important to determine when the RL agent is close enough to the goal, within a certain boundary of the optimum system state.
Robustness	Robustness of the digital replica to deal with noisy inputs or errors in the inputs.	Operating boundaries of the AI solution. Outside the operating boundaries it is difficult to guarantee properties on behavior.
Reliability	The digital replica should provide feedback when something goes wrong, for example when an input is given that does not meet the pre-conditions.	The AI solution should detect whether the system encounters unforeseen conditions. In such a case, mitigation should be performed. Examples are a restart, loading another model to deal with the problem, or raising an error message to a human operator.

Reproducibility	A deterministic digital replica makes it easier to reproduce results.	Important aspects include data labeling, document training of AI, and version control. Reproducibility also requires traceability linking experimental results to the used algorithm, data set, and settings.
Time to result	Digital twinning calculation time as part of the total training time	Training time of the RL agent depends on various factors including number of dimensions, features versus raw data, size of the state space, and available computer resources.
Scalability	The digital replica should be generic enough to enable learning of a valid zero-shot policy that covers other scenarios.	The training data should include sufficient variations to ensure that the trained RL agent can deal with different types of related systems. Important related techniques in the context of scalability are domain adaptation and transfer learning.
Explainability	Explainability is improved if the digital replica resembles the PT, having the same sub-modules/same architecture.	To what extent the RL agent can be explainable depends on the underlying technique. In general, this is not quite difficult. In the TEM case, we considered DQN, a value-function based RL using a neural network for function approximation. We used Q-values and the sum of the Q-values together with analyzing the action maps to explain the policy.
<i>Non-functional properties</i>		
Footprint	Extensive digital replica models and computations require significant resources.	The footprint of the RL agent is influenced by the number of training iterations required, the amount of training data and the required resources to run the RL agent once deployed.
Integrability	Integrability of the digital replica is affected by the needed resources, interface definitions, and required deployment approach.	Influenced by the software packages/plugin-ins being used for data handling/transformations and training the RL agent. Regarding data, different packages might be required depending on the type of pre- and postprocessing and whether the output type is continuous or discrete.
Maintainability	Understandability of the digital replica is crucial for maintenance. Maintainability can be improved by having a plug-and-play mechanism for new versions of digital replica sub-components, as well as having standard interfaces and connectors.	Understandability of the RL agent is an important aspect. Next to that, the software packages being used play an important role; whether they are maintained, do they have stable interfaces over versions, is there sufficient documentation.
Development cost	A complex and high-fidelity digital replica requires more effort, compared to a lower-fidelity digital replica.	The effort to develop an RL agent might be affected by where you must start. If there are similar solutions already available and have proven their value, then less experimental research might be needed.

6.5 Infrastructure

An important aspect of the ASIMOV solution is the required infrastructure. Especially, training an RL agent requires infrastructure that can store, transfer, and compute on big data sets. Once the RL agent is trained, the requirements on the infrastructure change. Executing a trained RL agent is typically much faster than training an RL agent.

Infrastructure should be considered in context of MLOps (Machine Learning Operations) that focuses on streamlining the development, deployment, and maintenance of machine learning models in production. General best practices around infrastructure, for example as introduced by Amazon Web Services [18], are also applicable when developing an ASIMOV solution.

7 Embedding in the organization

Although the ASIMOV consortium has been working mainly in the proof-of-concept phase, looking forward to productization and its implications seems beneficial. This is because introducing AI and digital twinning technologies have quite some impacts outside the technical domain. As the AI research field is booming, an enormous amount of literature, blogs, etc. appears daily, it is tempting to create an overview of the latest research results. We will not do this and focus on our own results and insights.

We use the five dimensions of the AI maturity framework [19] for organizing this chapter, as it zooms out from AI-technology and has the entire organization in its scope. This framework is also used as a basis for the questionnaire we sent out to the partners in the ASIMOV consortium. We report on the results in each section.

7.1 AI Maturity Framework

The AI Maturity Assessment Framework provides a means for organizations to reflect on their AI strategy along five dimensions; strategy, data, technology, people, and governance (see Figure 7.1). On each dimension, an organization progresses over five stages, each stage indicating a higher maturity; exploring, experimenting, formalizing, optimizing, and transforming.

Dimensions The following five dimensions are used to structure AI maturity of an organization:

- **Strategy:** The plan of action for achieving the desired level of AI maturity in the organization.
- **Data:** The data required to support specific AI techniques defined by the AI strategy.
- **Technology:** The technical infrastructure and tools needed to train, deliver and manage AI models across their lifecycle.
- **People:** The leadership practices as well as roles, skills and performance measures required for people to successfully build and/or work with AI.
- **Governance:** The policies, processes and relevant technology components required to ensure safe, reliable, accountable and trustworthy AI solutions.

Maturity stages Progress on each dimension is made following the following five stages:

1. **Exploring:** Exploring what AI is and what it can bring to your organization. The organization does not yet have an AI model or solution in production.
2. **Experimenting:** Experimenting with Proofs of Concept (PoC) and pilots. The organization is trying to put AI into production and can do so in limited ways only.
3. **Formalizing:** Moving from POC/pilot to an AI solution in production. Putting AI solutions into production still requires significant organizational work at this stage.
4. **Optimizing:** Scaling AI solution deployments efficiently as the number of deployed AI models increases. The organization is approaching a factory for model production.
5. **Transforming:** Transforming the organization itself using AI. The organization uses AI in how it operates across many critical areas of the business.

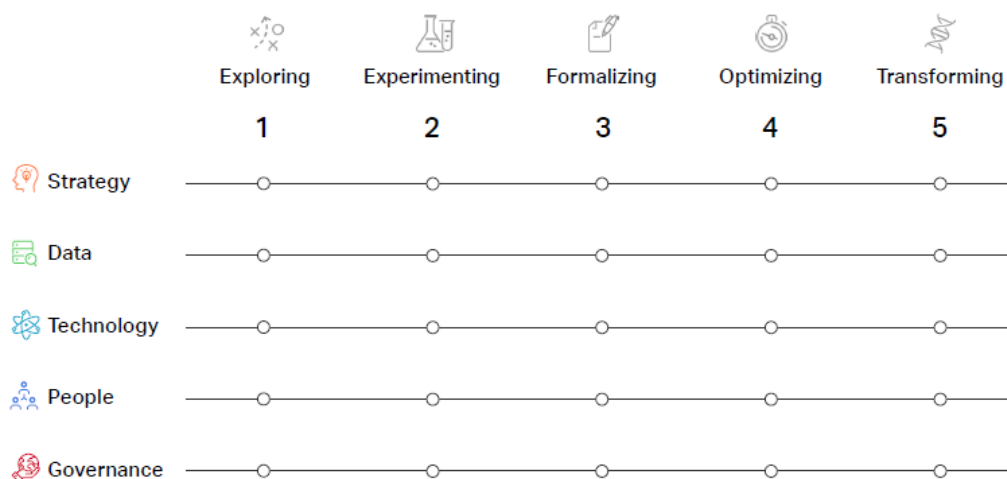


Figure 7.1 The five dimensions are ranked according to their maturity in the AI Maturity Assessment Framework.

The framework was initially developed by Element AI (in 2019-2020) based on a survey of around 200 organization leaders, mostly North American. Later, it was updated by Forum IA Quebec [19], together with Université Laval. By filling in the assessment tool related to the framework, organizations get insight into where they are standing on the dimensions, and how to advance in their AI strategy.

For the ASIMOV context, we extended the scope of the framework to also include digital twinning. We created questions that focus on either AI, digital twinning, or the combination. We also added a distinction between the progress stage at the start of the project, at the end of the project, and the desired stage to be reached in the future.

7.2 Strategy

7.2.1 Business

The drive towards a desired level of AI maturity in the organization must come from an expectation of AI having a high business value. To create business value with a DT+AI solution, organizations must carefully consider the embedding of the DT and AI technologies and the combined DT+AI solutions associated with it. Creating business impact requires a successful embedding of the solution along the dimensions as mentioned in the framework.

7.2.2 Business case

Naturally, the business of AI-enabled products or services is founded on providing customer value. This will generate revenue and profit. The AI strategy and DT strategy of a company should focus on finding the right markets and business cases for sustainable profit.

The value for customers can be expected in:

- **ease of use:** the AI can help in interacting with the product (e.g., explaining, automating, and supporting the interaction).
- **improved productivity:** automation promises faster of system use, and AI may automate in areas that were previously impossible. This leads to workflow efficiency, and contributes indirectly to uptime (e.g., less time spent on resolving optimization-related problems).
- **dealing with personal shortages:** many organizations face difficulties with finding and training highly qualified people and production workforce. AI can help to alleviate this challenge by automation.
- **effective workforce:** human resources can focus on interesting and valuable work.
- **new functionality:** AI and DT technologies may uncover valuable functionality (e.g., discovery of patterns in user generated data, recognition of product usage patterns, reasoning about planning).

7.2.3 Tactics for introducing AI and DT

As already mentioned in Chapter 3, the introduction of AI company-wide requires a sound strategy. Some of the tactics needed to be successful in the chosen strategy are shown here.

7.2.3.1 Showing results

A quantitative assessment is required to do early validation. Because of the purpose of the DT to provide useful training data, measuring 'how well the DT did' would be such a quantification. However, this can only be done once the RL agent has been trained using a digital replica and applied to the PT. This points to the advice to *close the loop* as quickly as possible.

Closing the loop can be done in two ways:

- *In an offline mode, where the RL agent has access only to the digital replica or a lookup table derived from the system data capturing the mapping from possible system states and actions to the new system states.*
- *In a real-time mode, where the RL agent has direct access to both the digital replica and the PT.*

Note that goals and therefore metrics differ depending on the development phase. In an early phase a low fidelity model and expected low performance (and other qualities) might be valid for the respective purpose. These will steer the developments in the right direction.

7.2.3.2 Scaling up

A key input for business decision adopting the proposed solution is the scalability of the solution. How does it scale along the business roadmap. Which product features can be offered when? To assess the scalability already at the proof-of-concept phase is important, but also difficult. Small-scale experiments and domain knowledge might help us here. An example: performing sensitivity analysis of interactions between an increasing number of parameters, in the case of system optimization by the DT+AI solution.

This approach is emphasised by the slogan: *Start small, scale fast*. Limited effort in the beginning but using full iterations to close the loop between DT+AI. Also, the validation with the physical system provides crucial information. This will confirm that the code base is working, and the project is working in the right direction. It is important to keep monitoring how the solution scales, not alone to have proof (data) when senior management might ask the question if they should 'pull the plug'. Of course, other supportive data must be gathered to counter these types of questions, but that is standard practice.

7.3 Data and Technology³

A systems architecture describes the dynamics and structure to build a system and selects the key technologies needed. In the current case, the focus is on two specific topics: data and AI technology.

The best practices for developing and using general AI systems are well-known:

- Data
 - Ensure that data is available for various purposes across the life cycle: installation, use, maintenance, and diagnostics.
 - Ensure that data is accessible, labeled, linked, of sufficient quality, volume, and proper representativeness, e.g., several instances of the CPS observed over time.
 - Ensure good tooling to support feature engineering, modeling, monitoring, and evaluation.
- AI technology
 - The AI-domain changes quickly nowadays, so AI-products can be developed also accordingly faster. This has consequences for AI-products in the field: they need to be able to integrate updates and upgrades.
 - For that an infrastructure must be in place (e.g., cloud/edge), containing infrastructure components for customizable configurations and workflows.
 - One can exploit the existing modularity and tiered architecture in the cloud/edge technologies. E.g., primitive components can be assembled in targeted ways to address less common and more specialized cases (see for instance [20])
 - Standardization plays an important role when products in the field must be maintained in a scalable way. This also holds for solutions in line with the ASIMOV architecture.

7.3.1 Architecture

There are special mechanisms required in the ASIMOV solution that influence the architecture in an uncommon way. Below we list a few of them:

- Training the RL agent using a digital model happens out of sync with the real system (the physical twin). This leads to the requirement for creating a representative model. The 'architecture' of the model itself must be such that it allows for synchronization.
- As the digital replica typically consists of different components, which may have different execution times, a mechanism for syncing must be in place. Next to that, a queueing system might be required in the training chain to deal with multiple instances with slower implementations.

³ We merged the Data and Technology dimensions as they are very closely related in the ASIMOV project.

- The digital replica structure may follow the internal structure of the physical system, but not necessarily. When it does, the hierarchical structure of the PT can be used to organize the structure of the digital replica.

7.3.2 Performance and qualities

One of the crucial qualities of the DT+AI solution of ASIMOV is the capability of generalization. Does training use digital twinning work out in the real world? This proves to be very difficult in industrial cases. The relevant complex relations hidden in the physical system data must be present in the digital twinning generated data, or the training will not be successful. For example, does changing a system knob have a linear impact on one element in the system state or does it also affect other system state elements. Therefore, utmost attention and effort should be spent on getting the digital replica 'good enough'.

This scalability of the model, the tuning of its fidelity, is supported by the architecture. This may be done via modular co-simulation:

- The modular co-simulation approach. Components are simulated together, but in their own environment. The result of each component is communicated to a central controller managing the start, timestep, etc. This has as advantage that accuracy settings, and even components, can be exchanged easily.

7.4 People and Organization

Lesser known is the big impact of 'AI' on the organization. The introduction of AI components in a product has impact on process: in the way-of-working as well as a huge impact on the product development process; on the organization: structure of departments and teams, the distribution of responsibilities, and the skills and development of employees.

7.4.1 Organizational aspects

Some aspects to consider when introducing AI and DT technologies:

- **Education:**
 - As AI and DT are relatively novel technologies, there should be knowledge build-up of DT and especially AI in the whole organization: senior management, product management, R&D, engineering.
 - There are successful channels to share information and educate on DT/AI-related topics, such as boot camps, dedicated training programs, and literature study (in office hours).
- **Community building:**
 - Build internally focused communities to share best practices, swap ideas, make connections for future collaborations (see for example [20]).
 - Alignment across the organization, cross-functionally, is an important aspect that must be initiated by senior management.
 - It is important to keep showing results to the organization (and being excited about them).
- **Engineering:**
 - An entirely new field may enter the organization on the development of AI systems: Systems Engineering for AI, for short: SE4AI. Next to that, dedicated development processes are needed when integrating AI components, requiring for instance knowledge on MLOps to develop and maintain AI components.
- **Roles and responsibilities:**
 - Roles and responsibilities will change, depending on the type: product teams, platform team, specialist teams, research teams [20].
 - An operating model must be developed: who manages AI resources, projects, solutions?
- **Job skills and talent strategy:**
 - Obviously, there is a need for domain experts on digital twinning and system modelling, AI, AI-leadership, AI-career paths within the company, etc. As mentioned before, many learning activities must be in place across the organization.

7.4.2 Starting development

Getting started with DT+AI solutions is not easy. We will mention here a few stumbling blocks we encountered and insights that we established.

Stumbling block: What the heck are you talking about?

At the start of many activities, a common vocabulary and understanding is lacking. Confusion, irritation and the like will appear, so act fast to resolve this issue.

Solution: in ASIMOV we have created a working group. This group provided a common understanding about terms and ideas. Eventually it worked so well this group provided a reference architecture that holds till the current day.

Insight: Distribute work

It is hard to work on the same artifact (model, code, experimental set-up). Splitting up the work is a common solution to this.

Solution: in the UUV use case the work was distributed into containers (Docker, separated execution environments). This allowed to work disjointly, but with well-defined interfaces, so the overall system could always progress. Also, the possibility to create mock-ups, from single source, and with one owner, made things easier.

Insight: Starting with collaboration

In the same light as the previous insight: collaboration is hard. Special effort must be put into this to make it work.

Solution:

- *When people from different backgrounds, or different companies must work together, they must connect 1-to-1, and face-to-face. Workshops are an important means to get everyone clear on the problem and what the plans are. In ASIMOV we experienced the great added value of meeting physically to get to know each other and establish good personal connections, especially to alleviate the challenges posed by working together mostly via Teams meetings due to the COVID pandemic and the physical distances between the different locations of the organizations.*
- *Have a shared generic demonstrator for collaboration with a clear end goal; e.g., the generic ASIMOV-in-a-nutshell demonstrator [21].*
- *It requires stamina to stick with a shared generic demonstrator, even when the use cases have mainly company focus.*
- *It is advised to have an onboarding program for new team members to build up domain knowledge required to develop an ASIMOV solution.*

Insight: Getting traction, improving status

It is important to communicate about developments and progress in the organisation and beyond (when allowed).

Solution:

Obviously, presentations in technology conferences, innovation events, and company-internal innovation markets are especially suited to convey the story, and get early feedback (acceptance in the company, realism of the solution, etc.). Having presentations and papers at conferences increases the status of the work considerably.

7.5 Governance

Disclaimer: in ASIMOV we have put only very limited focus on the dimension of governance. Therefore, we present here merely an overview of important aspects and some references to consider regarding governance.

Governance deals with the policies, processes and relevant technology components that are required to develop AI solutions that are safe, reliable, accountable, and trustworthy. AI governance includes

oversight mechanisms that address risks like bias, privacy infringement and misuse while fostering innovation and trust. This requires the involvement of a wide range of stakeholders, including AI developers, users, policymakers and ethicists, ensuring alignment with society's values.

Transparent decision-making and explainability are critical for responsible AI system behaviour. AI systems make decisions all the time, therefore it is essential for users to understand how decisions are made. This is important for holding systems/owners/manufacturers accountable and ensure that decisions are fair and ethical.

There are standards, frameworks and guidelines: the General Data Protection Regulation (GDPR) describes rules for protection of personal data and privacy; the AI innovation package by the EU Commission that supports startups and SMEs respecting EU values and rules; the Organisation for Economic Co-operation and Development (OECD) provides best practices and AI principles; etc.

At company level, there are numerous options to implement AI governance: Corporate AI Ethics Boards which oversee AI developments; dashboards that keep track of AI systems; metrics of the 'health' of AI models; performance alerts that notify responsible persons in case the AI system runs out of its predefined bounds; audit trails for accountability and reviews; etc.

7.5.1 Regulations and legal aspects

Regulations and legal aspects play an important role in governance. They should be considered from different perspectives. For example, regulations and legal aspects can be considered along the four BAPO perspectives [22]:

- Business: identify, analyze relevant regulations and associated restrictions
- Architecture: integrate mechanisms in the architecture to ensure legal and regulatory compliance
- Process: implement, execute processes in line with regulatory requirements
- Organization: build up and maintain knowledge on governance and legislation

The developments in this area are very rapidly progressing, keeping up is a challenge on its own. This indicates specialists are needed for this role. Given this situation, it is useful to consider the following recent developments.

Dutch Government

The AI Impact Assessment [23] is a tool utilized to facilitate discussions on AI systems. It explores obstacles in the data, the system, and the algorithms, taking account of applicable rules and regulations. The AIIA serves as an instrument for dialogue and for recording thought processes, thus enhancing, among other things, accountability, quality, and reproducibility. The AIIA is expected to result in a comprehensible document which clearly manifests the considerations underlying the decision to use AI in a project.

EU AI Act

The European Union is currently working on an AI Act to regulate systems that contain an AI component. The regulation is based on a classification system that determines the level of risk that an AI component could pose to humans or society. A higher risk means that there will be stricter regulations. The AI Act focuses mostly on rules concerning data quality, transparency, human oversight and accountability, as well as ethical considerations. A first position, the so-called "general approach" on the AI Act has been adopted by the European Council of the EU in November 2022 [24].

8 Conclusion

In this cookbook, we have gathered our advice to help you decide whether an ASIMOV solution with digital twinning and AI could be fit for your problem, and guidance to help determine whether there is a business case. If so, the cookbook provides pointers on how to get started. We have also given an overview of our experience in setting up an ASIMOV solution, focusing on the technical architecture, and the AI and digital twinning parts. We have also sketched the landscape on how to scale the initial solution from a prototype to a system-integrated solution and how to embed it in the organization.

Reflecting on our ASIMOV journey, we conclude that the approach was very promising at the start of the project but is also very difficult to realize. As the field of reinforcement learning is still in its infancy, there are few best practices available, and the tool and algorithm landscape changes very quickly. Next to that, there are not many industrial case studies available yet that have successfully integrated a solution with digital twinning and AI in a high-tech system, focusing on solving a control problem involving also hardware like sensors and actuators next to software. We have made very insightful first steps on the journey, but there is an exciting road ahead of us.

Acknowledgements

We would like to thank everyone in the ASIMOV consortium for their contributions to the ASIMOV cookbook. Especially, we would like to thank Bram van der Sanden, Richard Doornbos, Jan van Doremalen for the coordination, and Niklas Braun, Pieter Goosen, Lukas Schmidt, Ilona Armengol Thijs, Hans Vanrompay, and Michael Wild for their role in the core team. We would also like to thank Jan Willem Bikker, Pieter Goosen and Ilona Armengol Thijs for reviewing the document.

9 Terms, Abbreviations and Definitions

Best practice	A best practice is a method, technique, process, or activity that is considered to be the most effective and efficient way of achieving a specific goal or outcome.
Lesson learned	A lesson learned is knowledge or insight gained from past experiences, whether they are successes or failures. Lessons learned are used to inform future decision-making and actions to avoid repeating mistakes and to replicate successful outcomes.
Stumbling block	A stumbling block is an obstacle or challenge that hinders progress or causes difficulty in achieving a goal or completing a task. Stumbling blocks can be internal or external factors that impede or slow down the accomplishment of objectives.
Performance criterion	A performance criterion is a specific and measurable standard or expectation used to assess the quality, efficiency, or effectiveness of a process, project, or solution. Performance criteria are used to evaluate whether goals and objectives have been met.
Success factor	A success factor is a key element, condition, or variable that significantly contributes to the achievement of a desired outcome or the success of a project, initiative, or endeavor.
Artificial Intelligence	Artificial Intelligence refers to the development of computer systems that can perform tasks that typically require human intelligence.
Digital Twin	A digital twin is a virtual representation or digital replica of a physical object, system, process, or environment.
Reinforcement Learning	Reinforcement learning is a type of machine learning paradigm where an agent learns to make decisions and take actions in an environment to maximize a cumulative reward or achieve a specific goal.
AI	Artificial Intelligence
DT	Digital Twin
RL	Reinforcement Learning
UUV	Unmanned Utility Vehicle
TEM	Transmission Electron Microscope

10 Bibliography

- [1] ASIMOV-consortium, "D1.2 M18 Proof of Concept Demonstration and Evaluation electron microscopy," 2023.
- [2] ASIMOV-consortium, "D1.3 M18 Proof of Concept Demonstration and Evaluation of Unmanned Utility Vehicle," 2022.
- [3] S. Aheleroff, X. Xu, R. Y. Zhong and Y. Lu, "Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model," *Advanced Engineering Informatics*, vol. 47, p. 101225, 2021.
- [4] D. Bertsekas, *Reinforcement Learning and Optimal Control*, 2022.
- [5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2018.
- [6] A. Plaat, W. Kusters and M. Preuss, "Deep Model-Based Reinforcement Learning for High-Dimensional Problems, a Survey," *arXiv*, no. 2088.05598, 2020.
- [7] ASIMOV-consortium, "D2.2 Methods and Tools for Training AI with Digital Twin," 2024.
- [8] W. B. Powell, *Reinforcement Learning and Stochastic Optimization: A unified framework for sequential decisions*, John Wiley & Sons, 2022.
- [9] I. 22989, "Information technology - Artificial intelligence - Artificial intelligence concepts and terminology," 2022.
- [10] ASIMOV-consortium, "D2.5 Digital Twin Validation - Methods and Techniques," 2024.
- [11] ASIMOV-consortium, "D4.a ASIMOV Reference Architecture," 2022.
- [12] C. Gomes, C. Thule, D. Broman, P. G. Larsen and H. Vangheluwe, "Co-simulation: A Survey," *ACM Computing Surveys*, vol. 51, no. 3, pp. 1-33, 2018.
- [13] M. Arnold, "Stability of Sequential Modular Time Integration Methods for Coupled Multibody System Models," *Journal of Computational and Nonlinear Dynamics*, vol. 5, no. 3, 2010.
- [14] M. G. Kapteyn, J. V. Pretorius and K. E. Willcox, "A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale," *arXiv*, vol. 2012, no. 05841, pp. 1-16, 2021.
- [15] W. Pang, "6 Tips for Building a Training Data Strategy for Machine Learning," 6 9 2019. [Online]. Available: <https://www.kdnuggets.com/2019/09/6-tips-training-data-strategy-machine-learning.html>. [Accessed 9 2 2024].
- [16] ASIMOV-consortium, "D3.2-3.3 Architecture and technical approach for DT-supported AI-based training and system optimization," 2024.
- [17] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson and D. Fox, "Sim2Real in Robotics and Automation: Applications and Challenges," in *IEEE Transactions on Automation Science and Engineering*, 2021.
- [18] Amazon, "Well-Architected machine learning design principles," [Online]. Available: https://docs.aws.amazon.com/en_us/wellarchitected/latest/machine-learning-lens/well-architected-machine-learning-design-principles.html. [Accessed 9 2 2024].
- [19] Vitrine AI Québec, "Artificial Intelligence Maturity Assessment Tool," 11 2023. [Online]. Available: <https://vitrine.ia.quebec/en/maturity>.
- [20] Uber, "Scaling Machine Learning at Uber with Michelangelo," [Online]. Available: <https://www.uber.com/en-NL/blog/scaling-michelangelo/>. [Accessed 11 03 2024].
- [21] TNO, "'ASIMOV in a Nutshell' demonstrator," GitHub, [Online]. Available: <https://github.com/TNO/Asimov-in-a-Nutshell>. [Accessed 11 03 2024].
- [22] F. van der Linden, J. Bosch, E. Kamsies, K. Käsälä and H. Obbink, "Software Product Family Evaluation," in *Software Product Lines*, 2004.
- [23] Government of the Netherlands, "AI Impact Assessment," 02 03 2023. [Online]. Available: <https://www.government.nl/documents/publications/2023/03/02/ai-impact-assessment>. [Accessed 29 02 2024].
- [24] European Union, "Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts," 25 11 2022. [Online]. Available: <https://data.consilium.europa.eu/doc/document/ST-14954-2022-INIT/en/pdf>.
- [25] ASIMOV-consortium, *ASIMOV - Full Project Proposal*, 2020.

- [26] V. Francois-Lavet, P. Henderson, R. Islam, M. G. Bellemare and J. Pineau, "An introduction to Deep Reinforcement Learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, 2018.
- [27] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. L. S. Dunning and K. Kavukcuoglu, "IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures," *arXiv*, no. 1802.01561, 2018.