Reinforcement Learning Process Control in Powder Bed Fusion Additive Manufacturing

Stylianos Vagenas

Supervisor: Dr. George Panoutsos
Dr. Iñaki Esnaola

School of Electrical and Electronic Engineering
University of Sheffield

This thesis is submitted for the degree of
Doctor of Philosophy

December 2025

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are my original work and have not been submitted in whole or in part for consideration for any other qualification in this, or any other university.

Stylianos Vagenas
December 2025

Acknowledgements

I would like to thank Dr. George Panoutsos for his supervision during my research journey in the University of Sheffield. His guidance has been of substantial help and his work ethic has given me inspiration for my future endeavours. Moreover, I would like to acknowledge Dr. Iñaki Esnaola for his help and advice in the early steps of my research, along with Chris Smith and Nicholas Boone (Wayland Additive Ltd.) for our fruitful collaboration and their valuable insights.

Abstract

Powder Bed Fusion (PBF), a prominent metal Additive Manufacturing (AM) technique, is an original technique for producing 3D parts by adding material iteratively, on a layer-by-layer basis. In PBF, a heat source melts metallic powder, layer-by-layer, enabling the creation of potentially complex parts with tailored geometries. However, despite this advantage, PBF remains challenging to analyse and understand across multiple scales due to complex, nonlinear thermal phenomena and interactions. This complexity, along with the lack of control-oriented PBF models, hinders the development of closed-loop control systems. As a result, practitioners still rely on empirical openloop settings rather than feedback, which often leads to suboptimal, inconsistent builds.

Existing control attempts, based on control theory, have shown promise when applied to simplified PBF models with fixed control targets. However, as part geometries and PBF settings become more intricate, the required models may be inaccurate or even unknown for some aspects of the process, making traditional control methods challenging to design and implement. In contrast, data-driven techniques, such as Reinforcement Learning (RL), offer a more flexible alternative. RL is trained to derive optimal policies through trial-and-error interactions with the control environment, bypassing model assumptions. This flexibility, however, comes at a cost, since RL faces critical challenges: in RL training stability, marked by unpredictable training behaviour and high variance, and in constraint handling mechanisms, essential in safety-critical tasks.

This thesis, presented as a coherent collection of publications, aims to bridge the gap between RL algorithms and their practical use in PBF. Specifically, this thesis focuses on addressing the RL stability and constraint challenges in the context of PBF, enabling reliability and broader adoption. The goal is to establish safe and effective RL control in real-world PBF settings, ultimately unlocking the potential of RL in critical AM applications.

Contributions summary

The key contributions of this thesis include: (1) a new, stable RL framework for PBF that reduces training variance and improves control performance; (2) demonstration of stability and control performance analysis; (3) a new, constrained RL framework for PBF that enforces zero constraint violation during training and control deployment; (4) validation and assessment of the constraint framework and control robustness under different constraint settings and; (5) validation and control performance analysis of RL methods on new, advanced PBF models. These contributions are demonstrated within comprehensive RL control applications in a variety of PBF scenarios, including simple and more complex geometries, noisy signals and varying control targets. Through these efforts, this thesis offers substantial advancements in both RL methodology and PBF process control, setting the groundwork for trustworthy, intelligent manufacturing systems.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

This chapter presents a brief background on Powder Bed Fusion (PBF), Reinforcement Learning (RL) and how these intersect in the context of this thesis. This includes a description of the fusion process, the current challenges, the need for process control in PBF and why RL is a viable control choice. Moreover, this chapter includes a mathematical background on RL control and a review on important RL challenges. A summary of the conducted literature review is also presented on PBF modelling and process control attempts, in order to present the literature gaps to the reader and conclude to the objectives of this research work. Due to the nature of this thesis by publications, some of this information is to be repeated partially in the following paper publication chapters.

1.1 Powder bed fusion

Additive Manufacturing (AM), also known as 3D printing, is an innovative manufacturing method used to create 3D objects, typically by depositing material layer upon layer according to computer-generated models [1]. Compared to traditional, subtractive techniques, AM provides advantages such as the ability to fabricate components with intricate geometries and customised microstructures at a lower cost [2]. These advantages become even more important when working with metal. Traditional subtractive methods of producing metallic components often involve multiple steps, such as drilling and welding, whereas metal AM can fabricate equivalent parts, near net shape, in a single step, reducing waste and post-processing requirements.

One of the most popular metal AM techniques is PBF [3], a technique which utilises the material (e.g., titanium alloys) in metal powder form. The PBF process begins by converting a 3D model into a series of cross-sectional layers and storing it into a file. The resulting model file is then imported into the PBF machine using dedicated software. Before printing begins, key process parameters are selected and configured to ensure the quality of the build. Once this setup is complete, the PBF manufacturing process begins. The powder is swept onto the build platform by a recoating blade, formulating the first layer. A heat source selectively melts the powder on the build platform following a scanning pattern, generating a meltpool [4]. Importantly, the meltpool is the area at the heat source and material interface where metallic powder particles fuse to form a pool of melt metal, which subsequently solidifies once the beam advances to a different location, see [5] and [6]. Then, the build platform drops one level and the recoating blade sweeps powder on the build platform, formulating the second layer. This second layer is again melted by the heat source and the process is repeated

until the part is completed, see [7] and [8]. After the process is finished, the finished part is removed and cleaned. Any remaining or unused powder can typically be recovered and reused after undergoing appropriate preparation. Figure 1.1 illustrates the step-by-step sequence of the manufacturing process. Figure 1.2 shows the meltpool progress during scanning. Figure 1.3 illustrates a schematic of the PBF process with the example of an overhang structure, for which supports are commonly used in order to address heat dissipation issues. Finally, Figure 1.4 shows two final parts, produced by a PBF machine.

In general, the PBF process has grown rapidly in recent years due to its potential for aerospace and biomedical applications [2]. The two main techniques in PBF processes are Selective Laser Melting (SLM), utilising laser beam heat source, and Electron Beam Melting (EBM), utilising electron beam heat source.

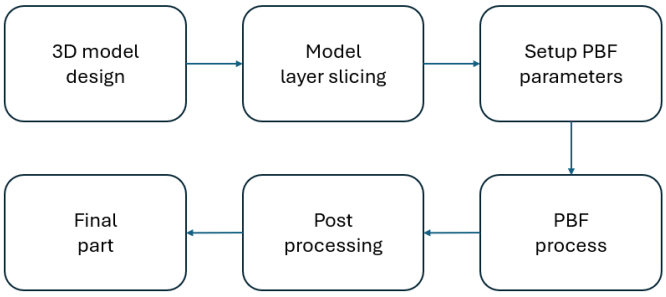

Fig. 1.1 Step-by-step representation of the manufacturing process.

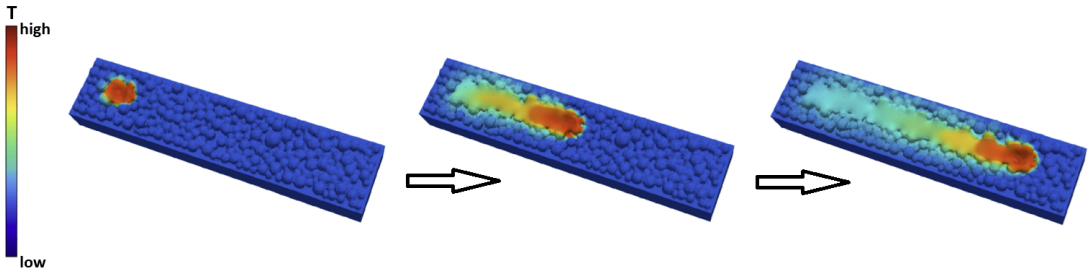

Fig. 1.2 Visualisation of the meltpool creation during PBF manufacturing, inspired by [6].

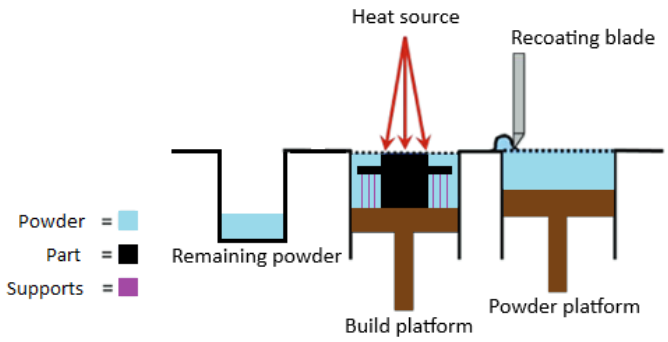

Fig. 1.3 Schematic of the PBF process, inspired by [4].

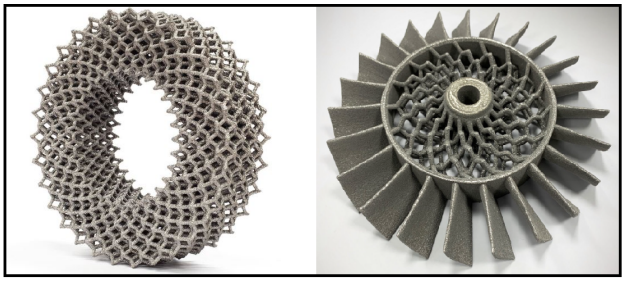

Fig. 1.4 Parts produced from a PBF machine, figure credit to Wayland Additive Ltd.

Despite its benefits, such as the ability to produce intricate geometries and reduce manufacturing costs, PBF has yet to fully realise its potential in terms of part reliability and final quality. One of the primary challenges lies in the occurrence of process-induced defects, which are often attributed to poor parameter setup and insufficient optimisation/control [3].

Unlike defects typically encountered in conventional manufacturing, the defects presented in this research work are mostly meltpool related. The meltpool plays a substantial role in the process, governing the consolidation of powder into solid material, and is therefore a critical focus for both process monitoring and research. It is commonly monitored using IR cameras to capture thermal profiles during fabrication [4]. Key meltpool characteristics, such as temperature and shape, are highly important, since they can influence the density and microstructure of the final part [9]. Some noteworthy defects (mostly meltpool related) occurring during PBF manufacturing include *porosity* due to lack of fusion, keyhole effect and gas entrapment, *surface roughness*, *distortion* due to large residual stress and *swelling*, as described in [10], [11], [12] and [13]. More specifically:

- Porosity: three types of pores dominate in PBF, which are lack of fusion pores, keyhole pores and gas pores. Lack of fusion pores are caused by insufficient amount of energy input in the powder bed. Increasing the input energy density is an excellent technique to minimise lack of fusion pores. On the other hand, keyhole pores are caused by excessive amount of energy input, which result in relatively large, keyhole shaped pores. Finally, gas pores are relatively small and they have spherical shape. They are caused by trapped gas during the melting process [10].

- Surface roughness: two are the main reasons for the formation of surface roughness. The first one is the staircase effect which is caused by the increasing layer number that is being deposited on the build platform. The second one is the attachment of partially melted powders to the external surface of the part being produced [11].

- Distortion: the main cause of distortion is residual stress. The produced parts are prone to a considerable amount of generated residual stresses, because of their inherent wide temperature gradients. It is mostly observed on overhang structures [12].

- Swelling: it is the phenomenon in which solid material can escape and end up on top of the powder melting plane. This resembles the humping effect seen in welding and results from surface tension forces affected by the shape of the meltpool [13]. This is mostly noticed in overhang structures within which conductivity is not sufficient.

The meltpool characteristics, hence the part's vulnerability to meltpool related defects, are affected by various parameters such as the heat source, the scanning speed, etc. [14]. Moreover, the metallic powder undergoes various state transitions, from metallic powder form to liquid and then rapidly to solid, dense metal. Finally, complex geometries can be vulnerable to heat dissipation issues, hence requiring special attention during manufacturing (e.g., overhangs). As a result, PBF comprises a highly complex process, which is arguably challenging to model accurately across all scales. This motivates further research and a literature investigation in order to evaluate current PBF modelling attempts and the state-of-the-art. The most dominant modelling approaches are found to be on SLM and EBM, including analytical modelling, combined with numerical techniques, such as Finite Element Method (FEM), and data-driven modelling, combined with Machine Learning (ML).

Some noteworthy analytical modelling attempts can be found in [15], [16], [17] and [18], in which analytical formulations are developed for processes such as SLM and EBM. More complex and integrated frameworks, such as those in [19] and [20], adopt multi-physics and multi-scale modelling by incorporating heat transfer, flow dynamics, grain structure evolution, and material behaviour. Moreover, efficiency-oriented techniques, such as the element birth/death method in [21] and the quiet element approach in [22], aim to reduce computational cost without compromising accuracy. Heat source modelling, an essential component in thermal simulations, is addressed in [23] and further expanded by [24] and [14], linking laser parameters with meltpool geometry. On the data-driven front, efforts such as [5] and [25] utilise ML and material databases to predict meltpool characteristics and geometric printability, demonstrating the potential of ML even with limited or imperfect training data. Finally, [26] provides a comprehensive overview of both modelling philosophies, highlighting the emerging prominence of data-driven methods as valuable solutions.

1.2 Powder bed fusion challenges

As the above literature summary confirms, PBF is a complex manufacturing process, and there is no accurate, integrated, computationally fast model to describe the relation between the process inputs and the final part's features across all scales. It is found that the more accurate the model is, the higher the computational cost that accompanies the model calculations. Hence, the most accurate models found in the literature are not control-oriented (computationally slow, complex relations) and the fast, control-oriented ones, follow simplistic modelling approaches that are not representative enough of a real-world process. Some noteworthy examples are the aforementioned works of [23], [24], [14] and [18]. In these works, the resulting models describe single-layer parts, and the correlation between the laser power and scanning speed with the meltpool temperature and geometry follows a straightforward calculation framework, not capturing the PBF process across multiple scales. Despite these challenges, their contributions in PBF control-oriented models are considered state-of-the-art and representative enough to provide good intuition. Hence, these models are used for benchmarking in this research work, whereas an important literature gap is spotted in 3D, control-oriented, computationally fast PBF models for simple and more complex (overhang) geometries.

1.3 Powder bed fusion process control

Process control is a critical task in industrial processes. Particulary, in PBF, it requires careful management to fully unlock the PBF potential for creating tailored microstructures, see [27], ensuring performance, enhancing surface and mechanical properties, and adapting to complex design requirements. Some of the most noteworthy PBF control attempts that are found in the literature are presented below.

Several process control strategies have emerged in the realm of PBF. Notably, in the work of [28], a closed-loop control method is developed and progressively refined to reduce meltpool temperature variation, improving part consistency and reducing defects. Other efforts, such as [29], [30], [12] and [18], with a feedback or feedforward controller targeting meltpool defects, are typical examples of process control based on meltpool metrics. More recent developments, including [31] with a Linear-Quadratic Regulator (LQR) control approach and the layer-wise closed-loop control strategy of [32], extend process regulation across layers. Moreover, a range of robust controllers, Proportional-Integral-Derivative (PID), Model Predictive Control (MPC) and Iterative Learning Control (ILC), have been implemented in studies such as the ones by [33], [34], and [35] to actively control meltpool temperature and geometry. Moreover, on the ML front, the work of [14] leverages ML and policy optimisation algorithms to fine-tune parameters such as scanning speed or laser power at the single-layer level. Finally, the more recent work of [36] utilise ML for porosity prediction and process parameter optimisation, validating the proposed method with experimental results, and unlocking the potential for process control. For broader context, general challenges and system requirements in AM control are discussed in the review works of [37] and [38], providing a comprehensive perspective on the field's trajectory.

Most of the attempts in the above literature are inspired by control theory, applied on single-layer (instead of actual 3D), analytical models and they implement control with fixed control objectives. However, as the need for more realistic simulations grows and the required part geometries become more intricate, the resulting models can be complex, noisy, less accurate, or even unknown. It is argued that in such cases, control theory techniques are challenging to implement, or even to design, since most of the dominant control theory techniques (such as PID and MPC) require strict signal assumptions and a model/plant for suitable design and control deployment. As an alternative, in order to address this challenge, data-driven ML control approaches are considered in this research work. More specifically, the focus is on the ML class of RL. Due to its flexible formulation, and model-independent nature, RL control approaches do not require model assumptions and are relatively simple to design. The benefits of RL in industrial tasks is discussed in [39] and an implementation example can be found in [14].

1.4 Reinforcement learning overview

The RL control framework belongs to the ML class of methods, providing a flexible control framework that reduces the need for strict model development. It is a data-driven, trial-and-error method that optimises control strategies through interaction with the process environment (training), driven by the goal of maximising a reward signal [40], see Figure 1.5. The RL framework consists of the following primary elements.

- The agent is the controller or the learner and it has the property of learning from its past experiences. It starts by acting randomly and, the more the training progresses, the more it is able to act and adapt automatically through learning.
- The environment is the world in which the agent lives and interacts. When an agent is in a state and performs an action in the environment, the environment returns a new state and the agent moves to this new state. From the RL agent's perspective, each interaction with the environment corresponds to a timestep, t, and occurs within a RL episode, which can be defined in various ways, e.g., infinite, finite with specific timesteps reached, finite with certain state observed, etc.
- The reward signal is the signal that defines the goal of the RL problem. High reward values translate to achieving the control target.
- The policy is the mapping between states and actions. It defines the learning way of behaving.
- The value function defines how good, in the long run, is for an agent to be in a specific state, in contrast to rewards which specify what is good in an immediate sense.

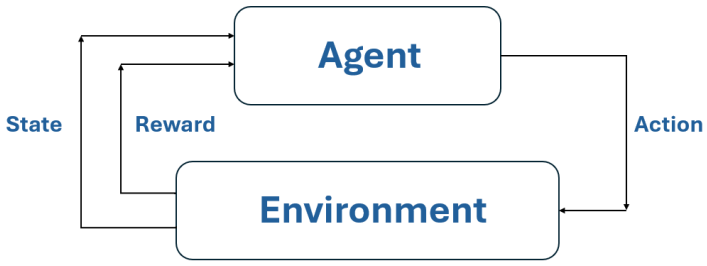

Fig. 1.5 Schematic of the RL framework.

In the RL framework, the process control problem can be formulated as a Markov Decision Process (MDP), which is a mathematical representation of a complex decision making process. According to [40], the MDP is defined by the tuple (S, A, r, Pr, γ) described below, while a diagrammatic representation of the RL training process is also given in Figure 1.6.

- S is the state space that describes the environment. In PBF, a state can include the meltpool temperature, the meltpool area etc.
- A is the action space of the agent. In PBF, an action can include the laser power, the dwell time, the scanning speed etc.
- r is the reward from the environment, after the agent takes action $a \in A$ at state $s \in S$. Intuitively, it is an inverse equivalent of the error signal between the achieved value and the target value in control theory techniques.
- Pr is the dynamics function $Pr(s', r|s, a)$, which denotes the probability of transitioning to s' and receiving reward r, if the agent takes action a at state s.

- γ is the discount factor or discount rate, which is a number between 0 and 1 $(0 \leq \gamma \leq 1)$ and determines the current contribution of future rewards.

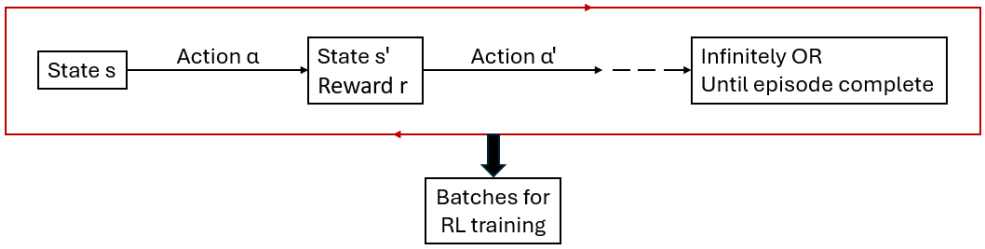

Fig. 1.6 Diagram of the RL interactions and sample collection for training.

A substantial measure in the RL framework is the cumulative discounted reward G, which is defined as

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \ldots \tag{1.1}$$

In the case of infinite MDPs, it is necessary that $\gamma < 1$ in order to keep G bounded. For finite MDPs, there are no further restrictions regarding the value of γ than $0 \leq \gamma \leq 1$.

Now, the value function $v_\pi(s)$ of a state s can be defined, since it is linked to the expected, E, cumulative reward following a specific policy π.

$$v_\pi(s) = E_\pi[G_t | S_t = s], \forall s \in S \tag{1.2}$$

$$v_\pi(s) = E_\pi[r_{t+1} + \gamma G_{t+1} | S_t = s], \forall s \in S \tag{1.3}$$

The above equation is a Bellman equation and it shows that the value function can be decomposed into two parts; the immediate reward and the discounted value of the successor state. The optimal value function $v_*(s)$ is defined as the maximum value function over all policies. It is the maximum possible total reward that the agent can get after leaving the state s. The ultimate goal for the RL algorithm is to find the policy π that gives $v_*(s)$.

$$v_*(s) = max_\pi v_\pi(s) \tag{1.4}$$

1.5 Reinforcement learning with function approximation

For discrete, low dimensional tasks, the necessary computation above is a simple process. For instance, if the MDP framework of interest consists of n states and m actions, the value functions can be stored in a $n \times m$ table and then temporal difference techniques or dynamic programming methods can be implemented to solve the MDP [40]. However, when the MDP framework of interest comprises a high-dimensional, continuous task, then the storage of the value functions is challenging and the computation time could grow prohibitively large. The most widely used method to address this challenge in RL is function approximation with artificial Neural Networks (NNs). NNs, by definition, consist of interconnected units (neurons) that pass information from one network layer to another network layer. A real-valued weight and bias are associated with each connection

between units. The units compute the weighted sum of their input signals and then apply an activation function to the result, in order to produce the unit's output. In the RL context, this method suggests that a parametric NN approximation $v_w(s)$ of the true optimal value function $v_*(s)$ is implemented, so that $v_w(s) = v_*(s)$, where $w \in R^d$ is a finite-dimensional weight vector and d is the dimensionality of w (the number of weights).

The vast majority of the current state of the art RL methods utilise NNs as function approximators for the value function and the developed policy, e.g., [41] and [42]. These methods belong in the actor critic class of RL methods (actor network and critic network), and they are dominating in the current literature, outperforming most of the other RL developed algorithms, both in discrete and continuous control tasks [43]. The NNs are denoted with $v_w(s)$ for the critic network, parameterised by w, that approximates the value, and $\pi_\theta(s,a)$ for the actor network, parameterised by θ, that gives the policy.

In the actor critic framework, the critic network is responsible for the approximation of the value function. The more the agent interacts with the environment (sample collection), the more informed and accurate the approximation is. The most commonly used objective function of the critic network is given in (1.5), with the goal of minimising the difference between the critic's current estimate, $v_w(s)$, and the target estimate based on experience. On the other hand, the actor network is responsible for policy development. The advantage function, given in (1.6), measures how much better or worse an action is, compared to just acting according to the current policy. The goal is to increase the probability of actions that lead to higher advantages. The most commonly used objective function of the actor network is given in (1.7).

$$J_{\text{critic}}(w) = \mathbb{E}_{(s,r,s')}\left[\left(r + \gamma v_w(s') - v_w(s)\right)^2\right] \tag{1.5}$$

$$A(s,a) = r + \gamma v_w(s') - v_w(s) \tag{1.6}$$

$$J_{\text{actor}}(\theta) = \mathbb{E}_{(s,a)}\left[ln\pi_\theta(s,a) \cdot A(s,a)\right] \tag{1.7}$$

In order to achieve these goals, the stochastic gradient descent/ascent optimisation method is usually utilised, depending on how one formulates the networks' objectives. During this training/optimisation process, the actor improves the policy based on positive advantages, hence the policy changes. Then, the critic updates the value function based on the new policy, and the advantage estimates decrease because of the improved new policy. This actor critic circle is repeated until advantages approach zero and the policy becomes optimal.

It is vital that the aforementioned metrics (rewards, advantages, objective function values) are measured during a RL training process, not only for troubleshooting but also for intuition. In practice, the most straightforward metric that one tracks to provide an assessment of the RL training progress is the reward function, as explained in the following section.

1.6 Reinforcement learning training assessment

The most commonly used tool for assessing the RL training process is the reward function graph. This graph consists of two axis, with the horizontal axis indicating the number of training timesteps, and the vertical axis indicating the achieved reward. A generally increasing behaviour of the achieved rewards is expected to be observed for successful RL training. For comprehensiveness reasons, Figure 1.7 is presented, in which three different RL training processes are given for the same control case study example. For this, episodic, control

case study example, each episode consists of 10 timesteps, and the reward is formulated in a way in which the maximum reward per episode timestep is 1 (hence, per whole episode, 10).

It is observed that the RL agents are all trained for 10000 timesteps (1000 episodes). The R1 agent demonstrates exceptional training, since it reaches increasing reward values in a consistent manner, converging to rewards close to the maximum possible. The R2 agent also demonstrates satisfactory progress, however, significant variance is observed in the early training stages. Moreover, it seems that it has not fully converged yet, since, despite the high reward values towards the end of the training, a slightly increasing behaviour in the last training steps can still be observed. Finally, the R3 agent does not demonstrate satisfactory training.

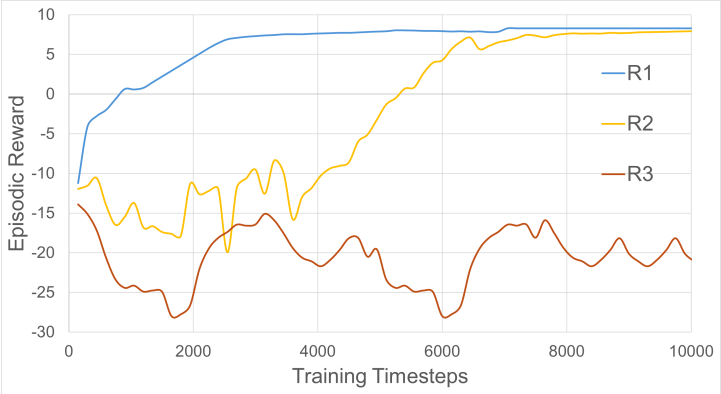

Fig. 1.7 Reward graph example of three different RL training processes.

Such differences among RL agents can be observed due to a variety of reasons. The most common reason is that R1, R2, and R3 agents are different RL algorithms with different designs and priorities. For instance, one algorithm may prioritise high rewards in an immediate sense, with high risk of sticking to local minima, whereas another algorithm may be more explorative. Another important reason for such differences lies on the chosen hyperparameters. Even in the case in which R1, R2, and R3 are the same RL algorithm, a difference in the chosen hyperparameters, e.g., learning rate, can lead to different RL training results. Finally, the reward formulation also plays a substantial role in the RL training progress. It is argued that the choice of the most suitable RL algorithm in a given control task, along with the most suitable hyperparameters and reward formulation, requires experience and good knowledge of the literature. In this research work, the focus is on the actor critic methods of Proximal Policy Optimisation (PPO) [41] and Soft Actor Critic (SAC) [42], as they are dominating in the current literature, see [43] for thorough assessment.

1.7 Reinforcement learning challenges

RL control algorithms, such as PPO and SAC, demonstrate high control performance in multiple case studies, see [43] for a thorough assessment. However, there are still important challenges that prevent RL from becoming a well-established control technique. First of all, the RL agent needs to be trained in order to derive a control policy, while there is no established way to predict/guarantee if the training is going to be successful. During training, the agent can present unpredictable behaviour and significant variance can be commonly observed in the reward signal, see [43] and [44]. The reasons behind these phenomena are still not fully understood, while effective solutions are needed in order to ensure a level of stability and repeatability in the

RL frameworks. Moreover, it is noted that there is currently no established way to apply constraints in the RL framework, while constraints can be a necessity for many tasks. The lack of constraint-aware RL frameworks comprises a vital reason for the absence of RL real-world implementations in critical tasks.

As it is also argued in [39], stability and constraints in RL control comprise substantial challenges in establishing the RL into the control mainstream. Hence, it is concluded that further research, new methods and new analyses are needed in the areas of stability and constraint satisfaction in order to create more trustworthy implementations of RL, which are required for critical applications. These two important challenges are defined as following.

- The stability in the RL framework is measured with regards to RL training variance. The improvement in stability is measured with regards to reduction in RL training variance, both overall and during the early, more uncertain, training periods, while achieving higher or at least the same reward values.
- The constraints under investigation in this research work are action-based (action constraints based on previous actions). Providing a zero constraint violation RL framework suggests that the imposed constraints are never violated during RL training and RL control deployment.

1.8 Research objectives

The aforementioned stability challenge has resulted in a number of works which have attempted stability guaranteed RL approaches, in various case studies, such as [45], [46], [47] and [48]. However, these works focus on environments for which the dynamic model is known or it is assumed to have deterministic behaviour. On the other hand the works of [49] and [50] are noteworthy attempts for RL implementation with stability in which the environment dynamics are unknown. These attempts provide good intuition for cases such as PBF, due to the modelling inaccuracy and interpretability challenges discussed in the previous sections. Regarding constraints, the works of [51], [52] and [53] highlight how RL can be compatible with constraint handling frameworks, and further investigation on PBF compatibility is made in this research work. Addressing these challenges and creating new RL paradigms with a focus on PBF implementation can be a substantial step for establishing RL process control in PBF. Hence, stability and constraints are chosen to be the main research directions of this research work, focusing both on the actor critic framework in general, but also, specifically, on PPO and SAC algorithms. The PPO algorithm is used in the first paper publication chapter, while the SAC algorithm is used for the second, third and final paper publication chapter. In every case in which these algorithms are used, careful tuning of the hyperparameters is implemented along with suitable reward formulations, always inspired by the existing literature and relevant studies.

Regarding the PBF application, as already stated, there is lack of control-oriented models available in the literature, and the existing control approaches focus on single-layer frameworks and simplistic approaches. Hence, the focus of this research is the application of the RL control methods in multi-layer, full 3D models, looking into features such as heat power, dwell time, meltpool geometry and temperature. These 3D models can either comprise expansion on previous works in the literature (currently 2D) or they can be derived by real-world experimental builds (collaboration with Wayland Additive Ltd.).

This research work aims to make a vital step towards surpassing the aforementioned barriers. It focuses on RL control for PBF processes, aiming to address the stability and constraint challenges by introducing new RL methods, applied in control-oriented PBF models. As a result, the following research objectives are pursued.

- RL: Investigation of current stability approaches in RL and potential stability guarantee techniques. Establish a novel RL framework that accounts for stability in PBF and apply RL control in a PBF model for intuition and assessment of the stability approach. Motivation: [49] and [54]
- RL: Investigation of the constraint needs in PBF and constraint compatibility with the RL frameworks. Establish a novel constrained RL framework, and apply control in a PBF model for intuition and assessment of the constraint approach. Motivation: [51], [52] and [53]
- PBF: Expand on an existing 2D model and establish a 3D PBF simulation model for benchmarking purposes. Motivation: [18] and [55]
- PBF: Creation of 3D PBF models based on real-world build data, developing both simple and more complex (overhang) geometries. Motivation: Wayland Additive Ltd.

1.9 Contributions

As previously mentioned, this thesis is a coherent collection of publications. Hence, in this section, the contributions of each of the following publication chapters are presented, covering both theory/methodology and application novelties. Unless stated otherwise, all contributions result solely from the research work of Stylianos Vagenas (supervised by Dr. George Panoutsos).

- Chapter 2 (Stylianos Vagenas and George Panoutsos. Stability in reinforcement learning process control for additive manufacturing.):

  -Understanding of stability in the RL framework and its potential practical challenges in PBF.

  -Investigation of the reward reformulation technique as a RL stability approach and application on a PBF platform.

- Chapter 3 (Stylianos Vagenas, Taha Al-Saadi, and George Panoutsos. Multi-layer process control in selective laser melting: a reinforcement learning approach.):

  -Further development of an existing 2D SLM model to a new 3D one (in collaboration with co-author Taha Al-Saadi), and implementation of process control on a simulated 3D SLM platform.

  -Demonstration of the benefits and limitations of a layer-wise control approach in 3D SLM, for simple and more complex control objectives (target tracking).

  -Reveal and reflect upon the benefits and limitations of a RL control approach, compared to traditional control theory based methods (control theory methods from co-author Taha Al-Saadi).

  -A new, stable RL framework is proposed and a demonstration of its benefits in the stability of the RL training and in the RL control performance is included.

- Chapter 4 (Stylianos Vagenas and George Panoutsos. Constrained reinforcement learning for advanced control in powder bed fusion.):

  -Investigation on existing constraint handling methods in RL and the constraint needs in PBF processes.

  -Introduction of a novel, constrained RL framework, with intensity tuning for the imposed constraint.

  -Application of a constrained RL control framework in PBF.

- Chapter 5 (Stylianos Vagenas, Nicholas Boone, and George Panoutsos. Bridging simulation and practice in additive manufacturing: reinforcement learning for electron beam melting control.):

  -Development of a real-world based EBM simulation platform (in collaboration with co-author Nicholas Boone) and implementation of RL process control.

  -Implementation of RL process control for simple and complex control objectives (varying target), including noisy signals.

  -Demonstration of the benefits and the limitations of RL process control on simple cuboid geometries, as well as more complex, overhang structures.

  -Assessment of auto-tuned, constraint-aware, RL control strategies, resulting to agents which are safe for real-world deployment.

Chapter 2

Stability in reinforcement learning process control for additive manufacturing

As previously mentioned, a major challenge in RL process control is regarding the stability of the RL training and the resulting RL control performance. This challenge poses substantial barriers to the application of RL in domains in which stability is a critical requirement. One such domain is AM, in which stable and reliable performance is required. Hence, the endeavour of RL control in an AM setting necessitates a deeper understanding of stability as a key criterion for successful RL deployment.

An attempt to enhance stability in RL control for AM is presented in [14], in which the authors propose a reformulation of the reward function by incorporating an additional stability term. This modification aims to encourage the RL controller to converge toward more stable policies. However, there is no established way to predict how the agent is going to react to this reformulation. A positive improvement in the RL training and performance is expected, but the reformulation not being effective is also a plausible outcome. In order to properly evaluate the effectiveness of this approach, this chapter tests the stability focused reward reformulation in an AM simulation, as shown in this chapter, section 2.4.3. Indeed, in practice, it seems that the reformulation of the reward function does not provide any stability improvement to the system. The findings reveal no statistically significant improvements in RL stability, highlighting the limitations of such heuristic based methods.

Consequently, this emphasises the need for more rigorous methodologies to address stability in RL. The literature summary and the stability discussion presented in this chapter, section 2.5, reveal that stability focused approaches grounded in Lyapunov theory, a cornerstone of control theory, can prove to be useful in the RL domain. This suggests promising outcomes based on Lyapunov-based frameworks, such as [49] and [54], to develop theoretically sound and practically effective solutions for stable RL control.

Note: the presented paper is slightly amended for the purposes of this thesis. The original paper has been peer reviewed and accepted in the International Federation of Automatic Control (IFAC) World Congress, 2023: Stylianos Vagenas and George Panoutsos. Stability in reinforcement learning process control for additive manufacturing. IFAC-PapersOnLine, 56 (2) : 4719–4724, 2023. Elsevier.

https://doi.org/10.1016/j.ifacol.2023.10.1233

2.1 Abstract

Reinforcement Learning (RL), as a machine learning paradigm, receives increasing attention in both academia and industry, in particular for process control. Its trial-and-error concept, along with its data-driven nature, make RL suitable for process control in complex tasks, in which the control task and framework can be formulated flexibly. However, there are still challenges that need to be addressed in order for RL to be introduced into the control mainstream, in particular for critical processes. A major challenge in RL is that there is no guarantee for robust, stable RL process control. This is a key impediment to RL implementation on tasks for which stability is an important requirement. Additive Manufacturing (AM) is an example of such process control task, since the very high complexity of the manufacturing process makes it suitable for RL process control, while stable performance is a necessity. However, one has to firstly understand performance and stability as a key requirement. In this paper, we reflect on stability approaches in RL and we investigate the stability requirements for AM. Our AM case study provides intuition and encourages further research for stable RL that would unlock potential for adoption and implementation of RL in AM applications. Research in the proposed direction would also have the potential for impacts to other process control sectors, in critical applications, in which appropriate utilisation of stable RL control could bring significant advantages.

2.2 Introduction

Additive Manufacturing (AM) is an original manufacturing technique for making 3D objects, layer upon layer, based on computer-designed models. In comparison with conventional manufacturing, AM offers the benefits of generating components with complex geometries and unique microstructures with reduced cost. These benefits have even greater impact when the manufacturing material is metal. Conventionally manufactured metallic parts may require a number of different processes, such as drilling, welding etc., whereas metal AM can produce the same parts with a single processing step, eliminating required cost and tooling. However, despite its advantages, AM is a complex process, including high levels of uncertainty, since it involves a variety of intricate underlying physical phenomena which are not yet fully understood, see [56]. Hence, the optimisation and control of the manufacturing process is a challenging endeavour which needs thorough investigation.

Feedback control methods which are based on control theory seem to be challenging for process control in AM. For instance, the utilisation of the Proportional-Integral-Derivative (PID) class of controllers seems to be one of the most popular approaches for process control. However, AM is a complex, multi-input-multi-output process which includes high levels of uncertainty. Thus, the PID formulation can be challenging for AM process control. The Model Predictive Control (MPC) class would also seem to be a valid option, since it can be flexibly formulated for multi-input-multi-output systems, it takes account possible constraints and it provides a predictive ability which is really helpful, especially in AM processes. However, MPCs (and PIDs) need a correlation or a model between inputs and outputs to apply control in the system. This correlation or model might not be known or accurate, and that is particularly the case in AM, since a comprehensive correlation has not yet been identified across all AM scales. The best available information is data, which show how the alteration of different inputs affect the outputs of the system. As a result, the combined challenges

that AM processes present, has led us to focus more on data-driven control methods which are capable of addressing these issues. These methods belong to the class of Reinforcement Learning (RL).

Despite the advantages that RL can present over the feedback control methods from control theory, there are still some challenges for RL to be established into the process control mainstream, see [39]. A major challenge is that the RL framework provides no stability guarantees. As previously stated, AM is a complex process with high levels of uncertainty, hence stability seems to be a highly important requirement. Thus, granted that the RL approach seems to be the most flexibly feasible for process control in AM, a stability guarantee approach needs to be introduced into the RL framework, to provide stable RL control in AM. The work of [57] was the first influential work regarding RL stability. After this work, some more recent attempts can be found in the literature, such as [49] and [47], which show that the Lyapunov approach is currently dominating. The Lyapunov approach for stability guarantee is a popular, viable approach in the control theory domain and it can be combined with RL, depending on the application. In the following sections, RL and the suitability of different stability approaches for RL process control are investigated, and particularly for AM.

2.3 Reinforcement learning process control

RL is a data-driven technique which is based on optimisation and decision making. It is about learning to map states to actions in an environment under uncertainty. The controller (agent) is not told which actions to take on a specific state, but instead must discover which actions to take by trial-and-error. This trial-and-error training is guided by the goal of maximising a numerical reward signal. In some interesting cases, except for the immediate reward, actions might also affect the future states, hence the future rewards. According to [40], these two features, trial-and-error searching and delayed reward, are the two most distinctive features of RL.

2.3.1 The reinforcement learning framework

In the RL framework, the process control problem can be formulated as a Markov Decision Process (MDP), which is a mathematical representation of a complex decision making process. According to [40], the MDP is defined by a tuple (S, A, r, Pr, γ) where:

- S is the state space that describes the environment.
- A is the action space of the agent.
- r is the reward from the environment, after the agent takes action $a \in A$ at state $s \in S$.
- Pr is the dynamics function $Pr(s', r|s, a)$, which denotes the probability of transitioning to s' and receiving reward r, if the agent takes action a at state s. The timestep of the process is represented as t, thus the state s' can be described as $s' = S_{t+1}$, the reward as $r = r_{t+1}$, the state s as $s = S_t$ and the action as $a = A_t$.
- γ is the discount factor or discount rate, which is a numerical value between 0 and 1 $(0 \leq \gamma \leq 1)$ and determines the current contribution of future rewards.

2.3.2 Formulating the reinforcement learning objective

A substantial measure in RL is the cumulative discounted reward G, which is defined as:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} ... \tag{2.1}$$

In the case of infinite (continuous) MDPs, it is necessary that $\gamma < 1$ in order to keep G bounded.

Now, the value function $v_\pi(s)$ of a state s and the action-value function $q_\pi(s,a)$ of taking action a at state s can be defined, since they are linked to the expected, E, cumulative discounted reward following a policy π. The policy π denotes the state-action mapping of the agent.

$$v_\pi(s) = E_\pi[G_t|S_t = s], \forall s \tag{2.2}$$

$$q_\pi(s,a) = E_\pi[G_t|S_t = s, A_t = a], \forall s, a \tag{2.3}$$

The equations above lead to the following:

$$v_\pi(s) = E_\pi[r_{t+1} + \gamma G_{t+1}|S_t = s], \forall s \tag{2.4}$$

$$q_\pi(s,a) = E_\pi[r_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a], \forall s, a \tag{2.5}$$

These equations are called the Bellman equations and they show that both the value function and the action-value function can be decomposed into two parts; the immediate reward and the discounted value of the successor state or state-action pair.

The agent's goal is to find the best possible solution in the MDP. Thus, according to [40], the optimal value function and action-value function have to be defined.

$$v_*(s) = max_\pi v_\pi(s) \tag{2.6}$$

$$q_*(s,a) = max_\pi q_\pi(s,a) \tag{2.7}$$

The optimal value function $v_*(s)$ is defined as the maximum value function over all policies. It denotes the maximum possible reward that the agent can get after leaving the state s. The optimal action-value function $q_*(s,a)$ is defined as the maximum action-value function over all policies. It denotes the maximum possible reward that the agent can get after leaving the state s, taking action a. If $q_*(s,a)$ is calculated, the optimisation problem is solved, because then the optimal policy is known. Thus, the ultimate goal for the RL algorithm is to calculate $q_*(s,a)$.

2.3.3 Reinforcement learning in additive manufacturing

In the literature, it seems that there are barely any attempts for RL process control in AM. Moreover, the existing attempts seem to be in premature levels, addressing mostly single-layer builds in simulation. For instance, the recent work of [58] proposed a RL approach for closed-loop control in a direct ink deposition application. This application included a single-layer build, and the proposed algorithm was tested in simulation. Their implementation achieved the desired deposition, with minimal height variation. [14], based on the

work of [23] and [24], created a simulation model that describes the temperature field that is created by a travelling heat source. [14] built a code that describes this model for a single-layer build and they tested the suggested RL algorithm in simulation. In the following section, the focus is on the work of [14], in order to gain intuition about RL process control in AM and stability issues. Finally, [59] focused on a wire arc application and proposed a RL framework for process control. It is found that a major contribution of their work was the implementation of multi-layer, real-world builds.

2.4 Motivation-case study

In this section, a case study for RL process control in AM is presented. This case study is based on the work of [14], which is correspondent to the powder bed fusion class of AM. The implemented control approach, varies the power of the laser beam to achieve the desired melt depth, following a predetermined scanning pattern. By this implementation, the credibility of the work of [14] is confirmed, along with the importance of their contribution regarding the simulation environment. The purpose of this case study, though, is to investigate the simulation results in terms of stability and gain intuition on the stability formulation for RL control in AM. In this AM case study, stability is addressed both in terms of the RL training, as well as in terms of the resulting melt depth variance.

2.4.1 Modelling

The model describes the AM heat source as a Gaussian distribution on a metallic surface. The heat source power P, the scanning speed V, and the scanning path, determine the resulting temperature T at the melt spot. The properties of the material, such as conductivity, specific heat, and melting temperature, are also defined and then, the simulation environment is ready to run. By varying the power of the laser beam, the temperature distribution around the melt spot is calculated, and the model gives the value of the melt depth as an output, using the mathematical model as in [14].

Regarding the simulated build, the model assumes a powder metallic plate of $1000\,\mu$m length, $1000\,\mu$m width and $300\,\mu$m depth, and a predetermined scanning pattern within this plate, see Figure 2.1. D_{melt} is the actual melt depth observed during the simulated manufacturing, while the desired melt depth is set to be $D_{target} = -55\,\mu$m. The scanning speed is assumed to be $V = 0.8\,\text{m/s} = constant$. For a more comprehensive view of the thermal model, see [14].

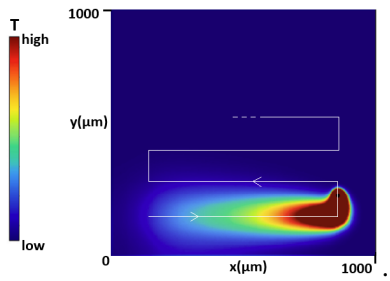

Fig. 2.1 Single-layer simulation build with a predetermined scanning path.

2.4.2 Methodology

By detailing the control elements in a MDP framework, the following are defined.

- Action space: The action space is a 1-dimensional continuous space. It is constructed by the power of the laser beam, $0\,\mathrm{W} \leq P \leq 500\,\mathrm{W}$.
- State space: The state space is a (9,10,10)-dimensional continuous space. It corresponds to the temperature distribution close to the melt spot.
- Reward function: The reward function per timestep is formulated as:

$$r = 1 - \left| \frac{D_{target} - D_{melt}}{0.5 D_{target}} \right| \tag{2.8}$$

  More specifically, the reward is an indicator of the percentage error between the actual melt depth and the target melt depth. This reward function can be reformulated by applying an extra term, in order to penalise for the difference between the maximum and the minimum melt depth observed during the layer build. The reward function, after the proposed reformulation, is defined as:

$$r_{reform} = r - \left| \frac{maxD_{melt} - minD_{melt}}{D_{target}} \right| \tag{2.9}$$

- Dynamics function: Unknown. The dynamics function do not need to be known or approximated, since a model-free control approach is taken in this case study.
- Discount factor: The discount factor is set to be $\gamma = 0.99$. Different values for the discount factor can lead to different results. However, the value of 0.99 is found to be dominating in the literature and it also gives the best control results in the specific case study.

The RL control method used in this case study is the model-free Proximal Policy Optimisation (PPO) algorithm, see [41] and [44]. The RL problem is formulated as episodic. Each episode consists of 101 timesteps, as dictated by the existing work of [14], and an episode is considered finished when the build of a layer is completed. The maximum reward that the agent can get per timestep is 1. Thus, the maximum reward per episode, i.e. per layer, is 101.

The role of the extra term in (2.9) is to avoid strategies that could cause "spikes in the melt depth that would otherwise be averaged out", see [14]. However, this is a heuristic approach to improve stability, since it does not directly affect the training process and the impact on the training or on the performance of the derived policy can not be guaranteed. Thus, while (2.9) seems like a rational formulation, it is important to highlight that this reward reformulation could encourage, but does not guarantee stability.

For further investigation, two classes of 5 test runs each are formulated, with the only difference being in the reward formulation. The training of the PPO agent is set at 4 million timesteps, for both classes. The first class of test runs utilises (2.8) (no stability term), and the second class, corresponding to the formulation of [14], utilises (2.9) (stability term).

2.4.3 Results

In Figures 2.2 and 2.3 the training process of the PPO agent is observed. It is argued that the performance of the agent is satisfactory in terms of reward, since it reaches rewards close to 101, which is the maximum reward that the agent can reach. The dense line is the mean return and the error bars correspond to the standard deviation observed in the experiments. In both experiments, the mean return and the standard deviation are calculated with respect to the 5 different test runs.

In Figures 2.4 and 2.5 the performance of the derived policy is observed, which is the result of the final training timestep. It is argued that the performance of the agent is satisfactory, but not ideal, since it generally reaches the desired melt depth, but there are some noteworthy levels of variance. The dense line is the mean melt depth and the error bars correspond to the standard deviation observed in the experiments. In both experiments, the mean melt depth and the standard deviation are calculated with respect to the 5 different test runs.

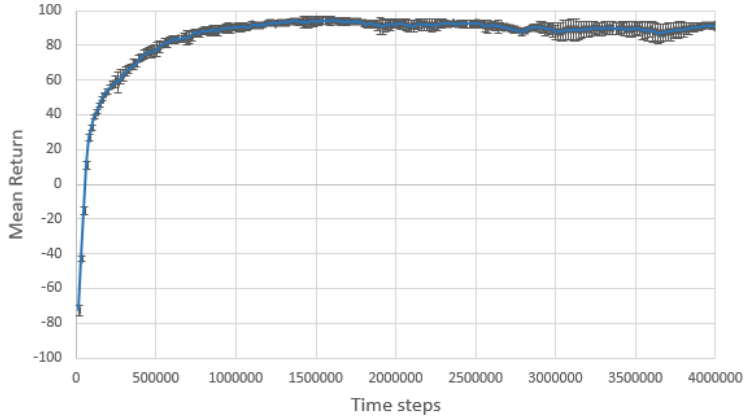

Fig. 2.2 Training curve of the RL agent, with the formulation of (2.8) (no stability term).

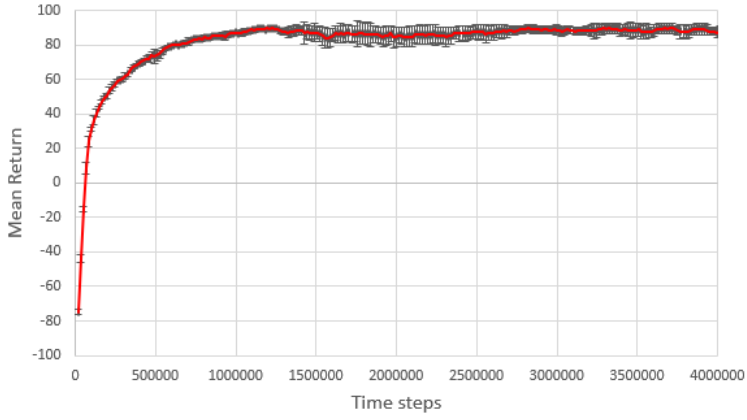

Fig. 2.3 Training curve of the RL agent, with the formulation of (2.9) (stability term).

The average value of the mean return of formulation (2.8) is 85.07 with a standard deviation of 18.88, while the average value of the mean return of formulation (2.9) is 81.82 with a standard deviation of 18.47.

Running a student's t-test with 5% significance level, a test is conducted to note if these results are statistically different. As expected, due to the different reward reformulation, the average values are statistically different, however, the standard deviations are not. This indicates that the formulation of (2.9) has no significant impact in the stability of the training curve.

The average value of the mean melt depth, from the policy derived from formulation (2.8), is -55.53 with a standard deviation of 3.91, while the average value of the mean melt depth, from the policy derived from formulation (2.9), is -55.63 with a standard deviation of 3.74. Running a student's t-test with 5% significance level, a test is conducted to note if these results are statistically different. Neither the average values of melt depths, nor the standard deviations are statistically different. This indicates that the formulation of (2.9) has no significant impact in the stability of the derived policy's performance.

The results of this case study show that the stability term plays no significant role neither for the stability of the training curve, nor for the stability of the derived policy's performance. As a result, a convincing conclusion on which formulation approach is more stable can not be reached. Hence, it is argued that this heuristic approach of reward reformulation is not sufficient for stability improvement of the system, and the motivation is to formulate more strict, mathematically rigorous approaches. The following section investigates stability in a more comprehensive way and reflects on attempts towards rigorous stability guarantees.

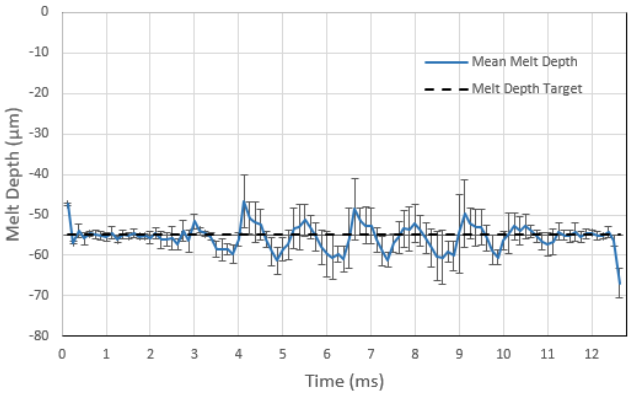

Fig. 2.4 The achieved melt depth of the derived policy, with the formulation of (2.8) (no stability term).

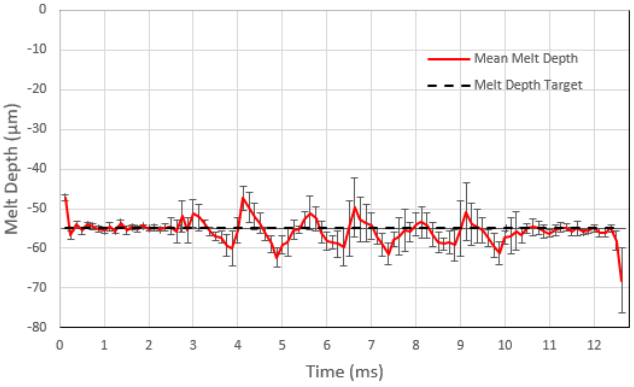

Fig. 2.5 The achieved melt depth of the derived policy, with the formulation of (2.9) (stability term).

2.5 Stability in reinforcement learning

Stability analysis in control systems is a research area that has been thoroughly investigated in the literature, e.g., see [60], [61] and [62]. However, in the RL control formulation, the stability analysis can be approached in different ways, which are not yet fully understood. This section investigates stability in a comprehensive way, making important distinctions among different aspects of RL and highlighting main instability factors.

2.5.1 Offline and online reinforcement learning

In order to gain more intuition about RL and stability, a distinction is made between offline and online RL. In offline RL, the RL agent is trained on a separate control environment, and not on the actual control system. This can be a simulation model or a replicate of the actual control environment. In this way, after the training is finished, the policy that best suits the stability criteria can be chosen and applied on the actual control system. In the case study of this work, this would mean that after the training, the best policy observed during the training could be chosen and applied on an actual powder bed fusion machine. A main drawback of this approach is that this policy is dependent on the accuracy between the training environment and the actual environment. On the other hand, in online RL, the RL agent is trained on the actual control system. As a result, the training curve actually corresponds to the real-time performance of the agent in the control system. In the case study of this work, this would mean that the return values of each episode, see Figures 2.2 and 2.3, correspond to the actual melt depths observed in this episode (via (2.8), (2.9)). A drawback of this approach is that the control environment could experience poor (and potentially, unsafe) policies from the agent either in the beginning of the training, or during unstable training periods. This is a key factor on why stability of the training process is a critical endeavour in online RL.

It is found that there are two major causes for instability in RL. The first one corresponds to noise in the calculations within the RL framework and the second one corresponds to the disturbance in the control environment.

2.5.2 Computational noise

As previously stated, the RL algorithm has to calculate the optimal action-value function in order to derive the optimal policy for the control problem at hand. In most cases, in which this calculation is not a straightforward process, neural networks are utilised along with gradient descent methods for optimisation. However, the fundamental presence of noise in the gradient estimators introduces instability in the training of the RL agent. As [63] and [64] suggest, a viable way of addressing this issue is the implementation of Stochastic Weight Averaging (SWA). SWA averages multiple samples from the optimisation trajectory and reduces the effect of noise in the estimators, since noise is eventually cancelled out. Although this method seems to be addressing the noise problem caused by the inner calculations of the RL algorithm, it does not address the instability caused by the disturbance in the control environment.

2.5.3 Environmental disturbance

Addressing the instability caused by the environment's disturbance is not a straightforward process. Each environment is described by its own characteristics and challenges. Hence, a different stability formulation is

needed for each different task at hand. This endeavour becomes even more challenging in RL when there is no model of the process. In the literature, there are some major implementations in RL, which attempt to provide stability guarantees in terms of disturbance rejection. It is found that the Lyapunov method is dominating, since most of the literature suggests either explicit Lyapunov approaches, or Lyapunov variants. For a thorough RL stability review, see [57], [65], [49], [66], [67], [47] and [45].

2.5.4 Lyapunov stability

The Lyapunov approach for stability guarantee is a generally accepted, sufficient approach in the control theory domain and the literature shows that it is compatible with RL in a variety of applications. For comprehensiveness reasons, Figure 2.6 is presented to elaborate more on the Lyapunov approach. Without loss of generality, the following approach holds in a n-Dimensional framework, where $n \geq 1$. Let Ω be a 2-Dimensional (2D) definite framework, with $x = (x_1, x_2)$. A target in the framework Ω is also defined as the stability equilibrium point $x*$. The Lyapunov function $V(x)$ is a continuous, differential function that is positively defined. More specifically, it is defined in a way that $V(x) > 0$ for every x in Ω, except for $x*$, where $V(x*) = 0$. The closer the vector x is to the equilibrium point, the closer the value of $V(x)$ is to 0. Thus, it stands that $0 < b < a$, see Figure 2.6. The goal is to follow trajectories with decreasing energy value of $V(x)$. Formally, $V'(x) = \nabla V(x) \cdot f(x) < 0$ needs to hold true for every x in Ω, where $f(x) = x'$ is the derivative of x. If these assumptions hold, then the equilibrium point $x*$ is locally asymptotically stable.

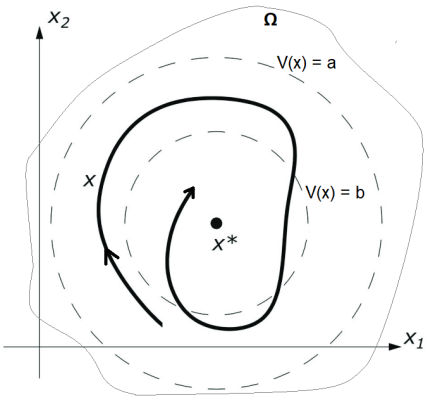

Fig. 2.6 Lyapunov function graph example for a 2D framework.

2.5.5 Additive manufacturing

The Lyapunov method has been widely used for stability evaluation and stable online control. It has been mostly combined with a known model, hence the Lyapunov function decreasing condition could be transformed into the expression of $\nabla V(x) \cdot f(x) < 0$ with known model parameters. However, this model might not be known or accurate. In a real-world application, a model describing the AM process across all scales does not exist. In the AM case study of this work, the model is used as a simulator, while the agent has no information about the model and relies explicitly on data sampling. Thus, a model can not be exploited in order to calculate the expression of $\nabla V(x) \cdot f(x) < 0$ with known parameters.

It is found that the work of [49] provides useful intuition for tasks of such nature. Their approach includes the construction of a Lyapunov function and approximation through sampling, without a dynamic model. More specifically, they suggest that the training of the agent should include a Lyapunov critic, which would evaluate the stability effect of taking a specific action in a specific state. In this approach, there are many assumptions that need to hold, along with a construction of a cost function. However, this is not a straightforward concept for AM, since it is still unclear if such assumptions can hold and a suitable cost function is needed. Hence, the application of such stability approach in AM needs further investigation.

2.6 Conclusion

In this work, we reflect on the stability challenges in the RL control formulation, along with the stability requirements of AM. To the best of our knowledge, this is the first work that investigates RL stability in the AM process control domain. The conducted literature review presents successful Lyapunov-based attempts in stable RL control applications, which provide good motivation for a Lyapunov-based attempt in RL control for AM. Moreover, the presented AM case study provides intuition about the stability challenges and our experiments highlight the need for rigorous stability guarantees in order to achieve satisfactory performance. Hence, we propose a Lyapunov-based RL approach for stable online process control in AM and we encourage more research towards this direction.

2.7 Funding

This study was funded by the UK Engineering and Physical Sciences Research Council (EP/P006566/1) and iCASE (EP/T517835/1), with contributions from Wayland Additive Ltd.

Chapter 3

Multi-layer control in selective laser melting: a reinforcement learning approach

Building on the direction suggested in the previous chapter, section 2.6, and examining the existing literature, Lyapunov-based RL methods are found to be effective for complex tasks in which stability is a critical requirement. The works of [49] and [54], which were mentioned in the previous chapter, provide a successful Lyapunov RL framework. However, the stability concept in their work, is based on strong assumptions on constraints and RL convergence, which are challenging to hold true. Arguably, this undermines the robustness of the stability guarantee claims presented in their work.

One cannot determine beforehand if the aforementioned assumptions and constraints are to hold true for a given control problem, especially when the environment dynamics are unknown. Moreover, even if they hold true, one cannot be certain if the proposed RL training will converge. RL is a method that utilises trial-and-error in order to understand the dynamics of the environment and control it. However, this requires sampling (state-action pairs) and this sampling is random. Hence, it is possible that the sampling will include state-action pairs that lead to a different converging direction than the desired one. In the case in which the RL agent is trained online, this phenomenon automatically disregards any stability guarantee claims. In the case in which the RL agent is trained offline, there is the luxury of choosing policies that lead to stable performance (if they exist) when the RL agent is deployed. However, even in this case, since the training occurs in a simulation or a replica of the actual control system, the stability guarantee depends on the accuracy between the training environment and the actual environment. As a result, due to the nature of the RL paradigm in both cases, there is no evidence to suggest that a theoretically guaranteed stable RL framework can be formulated in such environments with unknown dynamic behaviour. If such framework is to be formulated, it is argued that certain assumptions need to hold true, along with harsh methods leading to RL convergence, which would inevitably deprive the flexibility and adaptability that RL has to offer in comparison with control theory techniques.

Motivated by these limitations, this chapter introduces a novel, more flexible RL algorithm designed to enhance stability in a PBF control problem. Inspired by existing RL techniques and incorporating concepts from Lyapunov theory, this algorithm minimises a stability related cost function by integrating an auto-tuned penalty mechanism, as shown in this chapter, section 3.4.5. Compared to simpler reward reformulation

approaches, this method offers substantial advantages, while there is no need for additional assumptions to hold true, thus maintaining the flexibility and adaptability of the RL paradigm.

In addition to this key novelty point, this chapter comprises a substantial step towards more complete PBF models. A computationally efficient PBF simulation framework is introduced, representing the class of SLM manufacturing. Unlike the platform utilised in the previous chapter (single-layer), the developed simulation platform represents full 3D parts (multi-layer) capturing the relation between beam power and meltpool area and temperature. This simulation platform is used for benchmarking and evaluation of the proposed RL methods, as shown in this chapter, section 3.5.

Note: the presented paper is slightly amended for the purposes of this thesis. The original paper has been peer reviewed and accepted in the Journal of Intelligent Manufacturing, 2024: Stylianos Vagenas, Taha Al-Saadi, and George Panoutsos. Multi-layer process control in selective laser melting: A reinforcement learning approach. Intelligent Manufacturing, 2024. Springer Nature.

https://doi.org/10.1007/s10845-024-02548-3

3.1 Abstract

Powder bed fusion is an original additive manufacturing technique for creating 3D parts layer-by-layer. While there are numerous benefits to this process, the complex undergoing physical phenomena are challenging to analytically model and interpret. Hence, integrated and control-oriented 3D models are lacking in the current literature. As a result, the state of the art in process control for the powder bed fusion process is not as advanced as in other manufacturing processes. Reinforcement learning is a machine learning, data-driven mathematical and computational framework that can be used for process control while addressing this challenge (lack of control-oriented models) effectively. Its flexible formulation and its trial-and-error nature make reinforcement learning suitable for processes in which the model is intricate or even unknown. The focus of this research work is selective laser melting, which is a laser based powder bed fusion process. For the first time in the literature we demonstrate the benefits of a reinforcement learning process control framework for multiple layers (complete 3D parts) and we highlight the importance of stability during training. The presented case studies confirm the effectiveness of the proposed control framework, directly addressing heat accumulation issues while demonstrating effective overall process control, hence opening up opportunities for further research and impact in this area.

3.2 Introduction

Powder Bed Fusion (PBF) stands out as an innovative approach to metal Additive Manufacturing (AM), attracting considerable attention from both academia and industry. This manufacturing method, based on depositing layers of material using 3D computer designs, offers notable advantages compared to traditional manufacturing techniques, see [68]. PBF allows for the creation of intricate metallic components, see [69], with complex shapes and microstructures. In contrast to traditionally manufactured metallic parts that typically require multiple processes such as drilling and welding, PBF achieves similar results in a single process, leading to a decreased reliance on various tools, see [70].

PBF consists of two main manufacturing techniques referred to as Selective Laser Melting (SLM) and Electron Beam Melting (EBM), see [68]. In both SLM and EBM, an energy source, whether it be a laser or an electron beam, see [69], selectively melts the powder distributed on the build platform. This study focuses on the SLM process, as it is presently the most widely used and commercialised. As discussed in the following sections, existing SLM modelling efforts frequently offer a complex physics or data-driven representation of the real process. These versions, while realistic, are not suitable for applying advanced control due to their complexity and/or computational cost. Hence, there is a research gap in control-oriented, integrated, and computationally fast models, capable of establishing the relationship between desired process characteristics and controllable parameters across all relevant scales. Consequently, in industry, manufacturers resort to optimisation based on experience, without incorporating any active feedback process control, or merely incorporating simple feedforward and predetermined fixed control profiles that do not fully exploit the advantages of the process.

Simple part geometries, combined with simple control targets (e.g., constant temperature) could be addressed sufficiently with simple control methods. However, as part geometry becomes more complex, process and part models become more realistic and the control targets become more intricate, sophisticated control methods are needed that often require models and strict assumptions (e.g., on sensing methods). In SLM, such models (e.g., integrated, across scales) do not exist, or sensing and signal property assumptions cannot be satisfied. Using machine learning to control the process is an alternative, for example via the use of Reinforcement Learning (RL). RL offers a straightforward control framework, often implemented as iterative optimisation, for which model development and signal property assumptions can be less strict. For example, [39] discuss the potential of RL for industrial process control, and [14] demonstrate a RL framework for printing a single layer for a SLM process. In this study, for the first time in the literature, the control of a SLM process across multiple layers (complete 3D parts) is investigated via RL. The resulting framework is benchmarked against Proportional-Integral-Derivative (PID) control. This work reflects on the advantages as well as the limitations of RL and the necessity for stability in SLM process control, see [71], hence, the investigation is extended to include a new, stable RL variant.

The purpose of this research study is to develop and evaluate a process control framework based on a control-oriented 3D model of the SLM process. The 3D model represents the multi-layer SLM process, with particular emphasis on demonstrating - as an example - the challenges associated with heat accumulation among the layers as the number of layers increases. It is shown that the absence of a controller leads to heat accumulation issues, hence feedback control is considered. As a result of this investigation, a RL framework is proposed as a control method to adjust the power of the heat source based on process monitoring and feedback. The proposed methods are benchmarked against a carefully tuned PID controller. The main contributions and remarks of this work are the following:

- Further development of an existing 2D SLM model to a new 3D one, and implementation of process control on a simulated 3D SLM platform.
- Demonstration of the benefits and limitations of a layer-wise control approach in 3D SLM, for simple and more complex control objectives (target tracking).

- Reveal and reflect upon the benefits and limitations of a RL control approach, compared to traditional control theory based methods.
- A new, stable RL framework is proposed and a demonstration of its benefits in the stability of the RL training and in the RL control performance is included.

3.3 SLM modelling

3.3.1 The SLM process

SLM manufacturing machines, also known as laser PBF machines, utilise metallic materials (e.g., titanium alloys) in powder form. This metallic powder is stored on the powder platform and it is swept onto the build platform by a recoating blade. Then, the laser beam heat source selectively melts the powder on the build platform following a specified scanning pattern. Afterwards, the build platform is lowered and the recoating blade sweeps a second layer of powder onto the build platform. This process is repeated until the final part is completed, see [4]. Figure 3.1 presents a schematic of the SLM set up, including the aforementioned features.

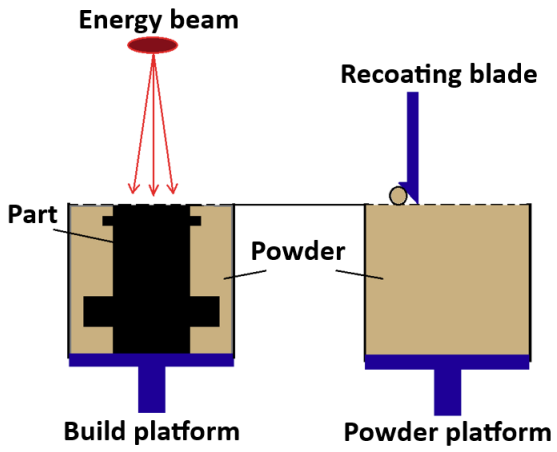

Fig. 3.1 Schematic of a SLM machine set up.

Extensive research has been dedicated to understanding the physics behind SLM processes, with a particular emphasis on the meltpool temperature, see [19] and [18], as this factor strongly correlates with the SLM part's quality. The meltpool refers to the region at the interface between the laser and the metallic powder, where the powder particles fuse together to create a pool of molten metal. Once the laser beam moves to a different location, the meltpool solidifies, see [69]. The meltpool's characteristics have great impact on the density and the microstructure of the component. Maintaining a uniform and consistent meltpool temperature and shape throughout the SLM build is crucial for preventing major defects, such as the formation of keyhole effects, see [10]. Heat accumulation among the layers of the SLM build would violate this goal of uniform and consistent meltpool temperatures and shapes. Hence, in this work, the heat accumulation challenge is selected as the primary objective to address.

3.3.2 SLM modelling efforts

During the manufacturing process, the metallic powder undergoes multiple transitions, from a powder form to a liquid state and then rapidly solidifying into a dense metal. These are very complex behaviours that make it challenging to create an integrated and precise model that accurately describes the relationship between the process inputs and the final part's characteristics analytically, at all scales. However, there have been significant efforts in the literature to develop models specifically focusing on the SLM manufacturing process. These modelling attempts involve combining analytical modelling efforts with numerical techniques such as the Finite Element Method (FEM), as well as data-driven modelling methods.

In a study conducted by [15], a technique was introduced to calculate the temperature distribution and stress within a single-layer build during the SLM process. The approach involved analytical physics formulations, and 2D FEM was employed for the necessary calculations. [21] employed an innovative simulation technique using 3D FEM modelling in their research. The simulation results indicated that "the heating and subsequent cooling to the ambient temperature occur within a few tenths of a millisecond of each other, thus suggesting that the irradiated spots are subject to rapid thermal cycles. These rapid cycles are associated with commensurate thermal stress changes." [21]. [17] analysed the modelling framework of PBF processes, with a specific focus on SLM. The authors introduced analytical equations for physics based modelling and utilised FEM to solve these models. [19] developed a comprehensive multi-physics and multi-scale framework that incorporated 3D modelling of heat transfer, flow dynamics, and considerations of grain size and microstructure. Numerical methods, including FEM, were employed to solve the equations of mass, momentum, and energy conservation. [18] created a control-oriented model to analyse the temperature and the dynamics of the cross-sectional area of the meltpool when scanning a part with multiple tracks (single-layer builds). Subsequently, a controller was designed with the objective of modifying the laser power. In the following sections, the work of [18] is used as a starting point and the 2D model is extended to a 3D one, in an attempt to establish a fully controllable SLM multi-layer model.

Regarding data-driven modelling approaches, [5] suggested the creation of a material database that could describe robust concepts within the SLM process. The developed model was capable of predicting the meltpool depth based on input parameters such as the power of the energy beam. It was found to perform sufficiently even when the training data was suboptimal, as long as appropriate physics filters were employed. [26] presented a compilation of analytical and data-driven modelling methods, highlighting the increasing prevalence of data-driven models. The study emphasised the significance of machine learning as a data processing technique to support data-driven modelling efforts. Finally, a more comprehensive review on modelling attempts can be found in the the work of [72].

3.3.3 Extending a 2D SLM model to 3D

For the model used in this study, the work of [18] is used as a template; the authors successfully created a control-oriented 2D model of a SLM process. The developed 2D model is based on building multiple tracks on a layer, following a back and forth scanning pattern. This model is extended, so that each time a new track is built, the model takes into account the heat accumulation due to the previous track, utilising the concept of a virtual heat source and applying the Rosenthal solution, as introduced in [73]. This virtual source concept is

shown in Figure 3.2. The material used in this work is Ti-6Al-4V powder and the manufacturing properties used are presented in Table 3.1.

In order to investigate the temperature behaviour within a layer, an example for a 4-track build is investigated, with 10mm length for each track, on a single layer. The layer consists of 800 points, which show the meltpool temperature history. The temperature observed is shown in Figure 3.3. As it is observed, the first track is built at a constant temperature, as there is no heat accumulation effect yet. From the second track onwards, temperature peaks are observed at the beginning of each track, due to the scanning strategy (beam passing next to recently scanned material) and the resulting overall heat effect of the previous tracks. Moreover, these temperature peaks tend to reach higher values as the building of the part progresses to further tracks built on the same layer. Given enough time, and a long enough track, the temperature within a track starts saturating towards the value that corresponds to the temperature that would have been achieved without the heat effect of the previous tracks.

Table 3.1 SLM model parameters

Parameter	Symbol	Value
Melting temperature (K)	T_m	1923
Ambient temperature (K)	T_a	292
Layer thickness (μm)	L_{th}	30
Scanning speed (mm/s)	V	800
Sampling rate (μs/point)	t_s	62.5

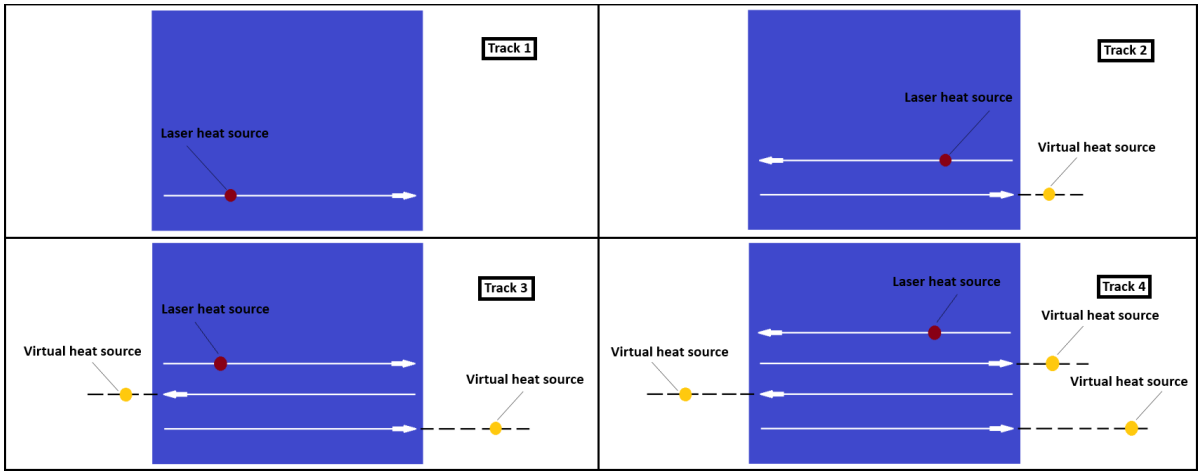

Fig. 3.2 Visualisation of the build of a layer, track per track. Example for a layer of 4 tracks.

Regarding heat accumulation, the Rosenthal solution is used, as introduced in [73], in order to estimate the heat effect of an already built layer to the next layer being built. In addition, a delay of five seconds is assumed between the end of one layer and the beginning of the next one, due to the time required for spreading the new powder after the completion of each layer. As a result, a 3D SLM model is produced, which does not only demonstrate how heat accumulates within each layer (2D), but among all the layers in height as well (3D).

In order to gain some practical intuition about the 3D SLM model's temperature behaviour, simulations corresponding to three different geometries are produced. The geometries are distinguished by the dimensions

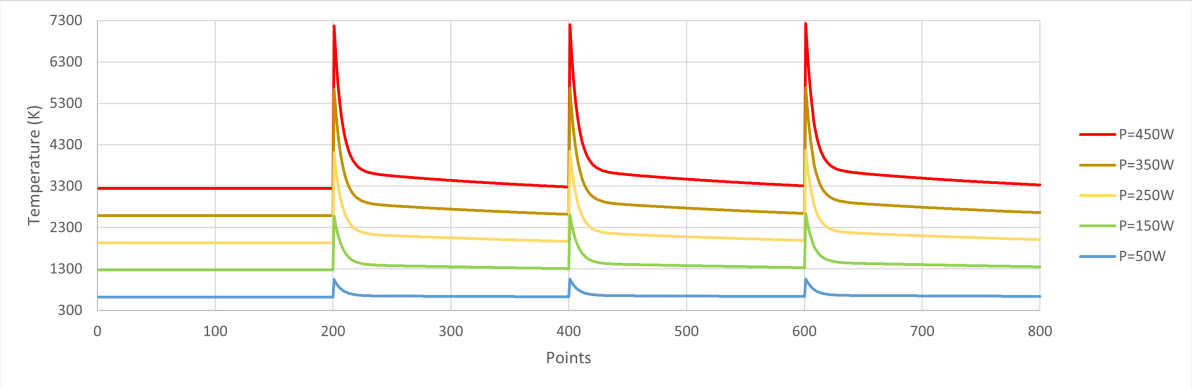

Fig. 3.3 Temperature history collected for different values of power, P, during a single-layer, 4-track build (10mm length for each track).

of the layer of the build, as *bigplate* (40 tracks, 10mm each), *rectangle* (18 tracks, 5mm each), and *thinwall* (4 tracks, 10mm each) geometry. The melting temperature of the material is 1923K. However, in the SLM model developed by [18], it is assumed that the steady-state temperature of the meltpool is a constant percentage higher than the melting temperature. As implemented by [18], the value of 20% above the melting point of the material is chosen in this study, which is 2308K. For each geometry, the laser power, P, is set to an integer value that approximately corresponds to an average layer temperature of 2308K. For the *bigplate*, the power is set as P=258W, for the *rectangle*, the power is set as P=198K, and for the *thinwall*, the power is set as P=280W. As seen in Figures 3.4 and 3.5 the average layer temperature of the first layer in all cases is approximately 2308K. As the build progresses and more layers are added, the average layer temperature increases because of heat accumulation. In all three scenarios, as many layers as necessary are simulated for the heat accumulation among the layers to reach a saturation point, so that one can appreciate the significance of the geometry of the build in heat accumulation issues. It is observed, that the *bigplate* geometry's heat accumulation saturates after 25 layers, the *rectangle* geometry's after 100 layers, and the *thinwall* geometry's after 200 layers. Moreover, it is measured that the average layer temperature difference between starting layer and saturation layer in the *bigplate* geometry is 10K, while in the *rectangle* geometry is 36K and in the *thinwall* geometry is 115K. Hence, it can be concluded that the *thinwall* geometry comprises the most challenging geometry regarding layer-wise heat accumulation issues. Therefore, the *thinwall* geometry is selected for the following process control case studies, in an attempt to clearly demonstrate the adjustment of the power accordingly on each layer, with a layer-wise control approach.

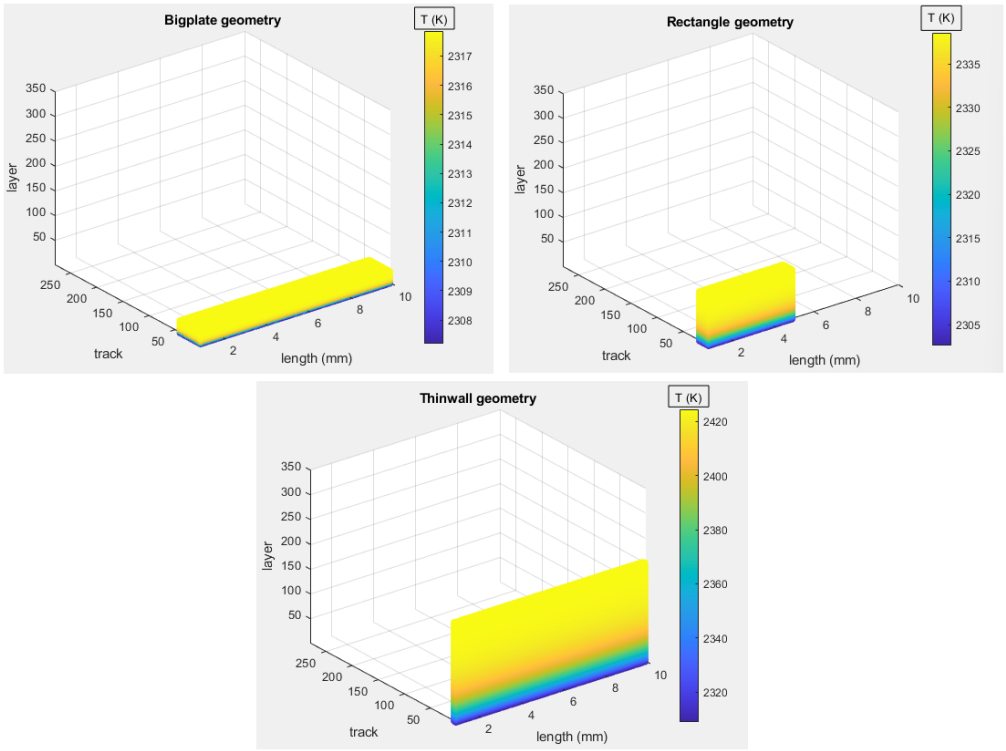

Fig. 3.4 Colormaps for visual representation of heat accumulation among the layers. Each point is denoted with the average temperature of the layer in which it belongs. Axes are in scale so that visual dimension differences correspond to actual dimension differences among the three geometries.

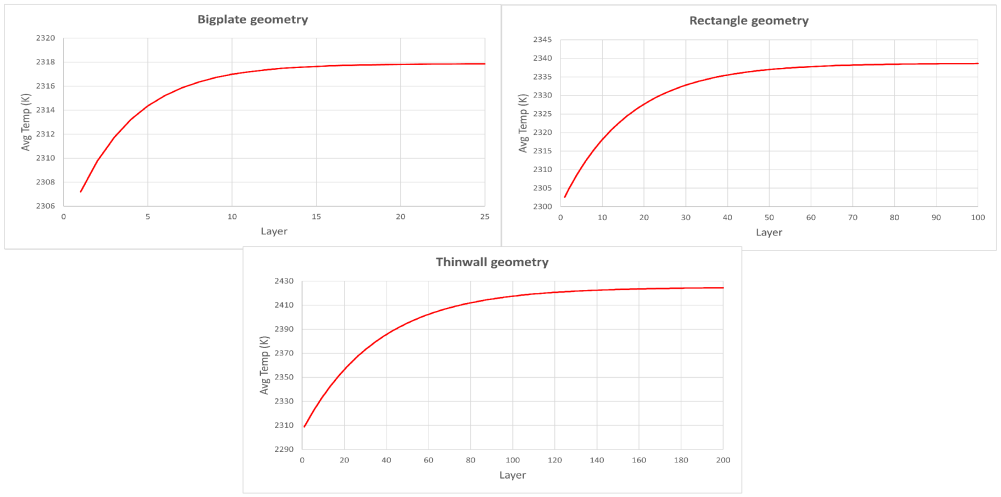

Fig. 3.5 Average layer temperature graphs for representation of heat accumulation among the layers. These graphs correspond to the respective colormaps in Figure 3.4.

3.4 Process control

3.4.1 The need for feedback control in PBF

Process control is a critical task in relevant industry sectors, see [74]. Within the PBF context it requires special attention to enable the process to realise its potential in terms of achieving bespoke microstructures, ensuring stability (e.g., for certification of aerospace parts), see [71] and [27], optimising surface and mechanical characteristics in complex designs etc. In this work, the focus is on process control techniques that are based on control theory, as well as on data-driven approaches in order to determine the benefits and the drawbacks for each class of methods. Control theory methods such as PID and feedforward control are the most popular approaches in industrial process control, mainly due to the simplicity of implementation, see [39]. The fundamental principle of the PID control, see [75], involves measuring the difference between the actual and desired system output signals. The PID controller then takes an action in order to minimise this difference, based on the dynamic characteristics of the error signal (K_P, K_I and K_D terms in the control law). These are well-established, interpretable methods that can be applied in a large variety of tasks. When addressing a simplified model of PBF, in which the complexity is low and the control target is simple, these theory methods seem to be the obvious choice as a first attempt to apply process feedback control in PBF. However, when the part geometry and underlying models become more complex and the control target is not as straightforward (e.g., multiple target tracking), simple control theory methods would not perform well in PBF (e.g., to counteract disturbances, signal delays, process drifts etc.). In this case, the most valuable information is monitoring and process data, in order to identify the relationship between the control system's inputs and outputs. Hence, data-driven approaches and feedback control methods could help alleviate some of the aforementioned challenges.

3.4.2 Process control in PBF

In this section, attempts at process feedback control for PBF in the literature are appraised. [29] presented a feedback control method based on a LabVIEW virtual instrument for the EBM process. They altered the process temperature, in order to achieve a desired grain size. The authors used a virtual instrument to apply control with feedback obtained via IR camera images and layer information decoded from calculations made by the virtual instrument's loop iterations. However, no systematic feedback control law was presented. [76] presented an approach of a P-controller which enabled fast control of the meltpool temperature. This approach reduced the deviation of process temperature by up to 73%, which led to more stable conditions in the meltpool, in comparison with constant laser power strategies. Based on this approach, [28] presented an improved method using a model-assisted version of the earlier attempt that reduced the deviation of process temperature by up to 90%. [30] utilised a PID control system for overhang structures, with a high-speed camera installed on the SLM machine. The control system was represented as a single-input-single-output system. [18] introduced a feedforward controller to address overheating issues and keyhole effects in SLM. The purpose of their controller was to maintain the meltpool cross-sectional area at a constant level throughout the building process. [31] introduced a control-oriented thermal model of a multi-layer SLM process and proposed an in-layer Linear-Quadratic Regulator (LQR) to track temperature. [32] were one of the first teams to attempt a layer-wise control strategy in SLM. They could effectively apply control to limit geometrical

defects due to overheating. [35] developed a robust controller, inspired by iterative learning control and online feedback optimisation, which altered the laser power in order to stabilise the inter-layer temperature. In general, the recent work of [38] covers control requirements in AM, for the reader to gain a broader perspective.

The aforementioned control techniques are either too simple to be effective at scale (e.g., PID) when part and process complexity increases, or require tuning which is not trivial (e.g., LQR) and strong assumptions (e.g., regarding disturbances, process drift), as discussed in the introduction section of this study. On the other hand, RL offers a data-driven control framework, for which learning algorithms are used to create an effective control policy. There seems to be still little evidence of RL based process control in PBF processes. State of the art RL attempts are still in a preliminary development phase, focusing mostly on single-layer parts or applied in other AM processes (e.g., blown powder), but not yet in PBF. In PBF, [14], influenced by the contributions of [23] in process modelling, and [24], created a simulation model that can track the temperature history that is created by a travelling laser beam, on a single-layer part, and implemented RL control to achieve a desired meltpool depth. Building on this work, [71] replicated the above results and demonstrated the need for stability in the behaviour of RL process control in PBF.

The above literature review shows that there are mostly simple feedback and PID based approaches for PBF process feedback control, while data-driven approaches based on RL have been only recently investigated in a preliminary fashion. Moreover, there is no evidence in comparing control theory with data-driven techniques, particularly in multi-layer PBF environments. Hence, benchmarking of these methods is needed, and a comparison between control theory and data-driven techniques is to provide substantial intuition on multi-layer PBF control.

3.4.3 Reinforcement learning overview

RL is a data-driven iterative optimisation approach centred around interactions with an environment/process. The controller, also known as the agent, serves as a decision maker, consistently learning and refining its control actions. Unlike being explicitly instructed which actions to take in specific states, the RL agent must discover the most effective actions through a process of trial-and-error, see [40]. This exploration is guided by the objective of maximising a numerical target, the reward.

A high-level diagrammatic representation of RL is shown in Figure 3.6. The RL sequence starts with the agent being in an initial set of states, s_t, with t denoting the corresponding timestep. The agent takes an action, a_t, in the environment's initial states, and the response of the environment is fed back to the agent with the form of a new set of states, s_{t+1}, and a corresponding reward, r_{t+1}. For example, within the context of PBF, the action could be the laser power of a SLM process, while the reward could cover maintaining a constant layer temperature.

The most popular and efficient approach, in terms of process control in RL, is the actor critic class of methods, see [41] and [42]. In Figure 3.7, a generic structure is shown for the control framework. In the actor critic class of RL, the agent can be formulated as a pair of neural networks, each of which has a very specific role to play. The neural networks are function approximators (policy function and value function respectively) and are denoted with $\pi_\theta(s,a)$ for the actor network, parameterised by θ, that approximates the policy distribution, and $v_w(s)$ for the critic network, parameterised by w, that approximates the value. The

magnitude of the update step that the neural networks go through during the learning process is dependent on the learning rate, denoted by α for the actor network, and β for the critic network.

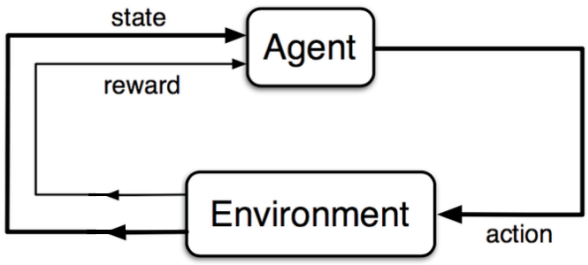

Fig. 3.6 The RL paradigm.

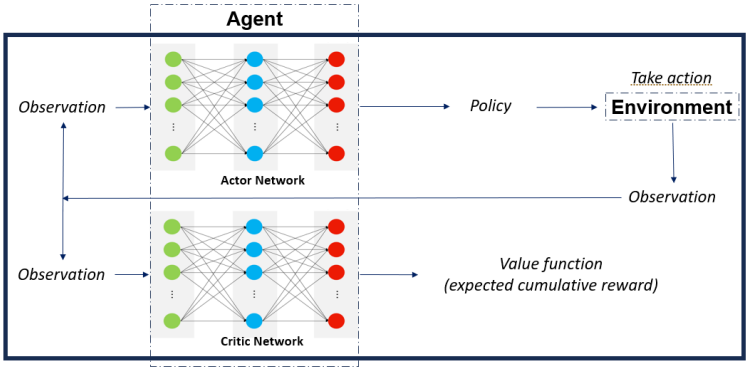

Fig. 3.7 The actor critic framework.

3.4.4 Soft actor critic method

The Soft Actor Critic (SAC) algorithm, introduced by [42], is a model-free actor critic method within the RL domain. It aims not only to maximise expected rewards but also to maximise entropy (via entropy regularisation), which enhances the exploration of control actions. Considered as a state of the art RL algorithm, SAC has demonstrated exceptional performance in continuous control benchmarks, see [43]. It employs an experience replay buffer to store previously collected agent-environment interactions, utilising them to enhance sample efficiency. The SAC algorithm consists of an actor neural network responsible for policy updates, determining action selection, and two critic neural networks responsible for evaluating the quality of the taken action. It balances exploration and exploitation, using an entropy coefficient (or reward scale) and this hyperparameter is found to be the only hyperparameter that needs to be tuned for SAC to perform. For this work, the SAC algorithm is used, as in the *stable-baselines3* platform by [44].

3.4.5 Proposed method: adaptive weighted actor critic

Despite the benefits of the actor critic framework, stability remains a key challenge in RL process control, see [39] and [71], as there are no performance guarantees. In an attempt to introduce more stable RL frameworks

for PBF, for the first time the Adaptive Weighted Actor Critic (AWAC) algorithm is introduced. AWAC is a model-free actor critic method which aims to improve the stability of the agent's training. It utilises an auto-tuned signal from the environment, which guides the agent towards its goal while stability is accounted for.

In the *advantage actor critic* learning process, see [77], a metric called *advantage* is defined, this is $A(s,a)$, as presented in (3.1). The actor network's update rule is presented in (3.2), while the critic network's update rule is presented in (3.3). The factor γ is a numerical value, such as $0 \leq \gamma \leq 1$, and determines the current contribution of future returns. The value of this discount factor is usually set empirically as $\gamma = 0.99$.

$$A(s_t, a_t) = r_{t+1} + \gamma v_w(s_{t+1}) - v_w(s_t) \tag{3.1}$$

$$\Delta\theta = \alpha \nabla_\theta (ln\pi_\theta(s_t, a_t)) A(s_t, a_t) \tag{3.2}$$

$$\Delta w = \beta \nabla_w (v_w(s_t)) A(s_t, a_t) \tag{3.3}$$

At the start of the process, the agent has an initial set of states, s_t. The critic network approximates the corresponding value, $v_w(s_t)$. The agent takes an action a_t in the environment, sampled from the actor network $\pi_\theta(s_t, a_t)$. A new set of states, s_{t+1}, with the corresponding reward, r_{t+1}, are fed back to it and the critic network, yet again, approximates the corresponding value $v_w(s_{t+1})$. Having this information, one can now calculate $A(s_t, a_t)$ and the update steps $\Delta\theta$ and Δw of the neural networks. The aim of this learning process is for the agent to be able to choose actions that lead to higher returns.

The main idea behind AWAC is the construction of a cost function, c_{t+1}, in which stability is accounted for, just as the reward function relates to performance. This cost function is a positive defined function and its formulation depends on the task at hand. This cost function is used to design a discount for the calculated advantage, $A(s_t, a_t)$, so that the agent is guided to perform in a stable manner. The novelty of the proposed approach is within the way this discount is designed.

In the proposed framework, an extra stability network, $d_z(s)$, is introduced, parameterised by z, which approximates the expected cumulative cost, as shown in Figure 3.8. In this way, a disadvantage, $D(s_t, a_t)$, is calculated as presented in (3.4). The role of $D(s_t, a_t)$ is to guide the agent towards stability. However, it should not subsume the role of $A(s_t, a_t)$ that guides the agent towards good performance. Hence, the impact of $D(s_t, a_t)$ is weighted by a factor, ω, with $0.1 \leq \omega \leq 0.5$, such as the new, total advantage, $A'(s_t, a_t)$, can be calculated in (3.5). As a result, the update steps for the neural networks can be calculated in (3.6), (3.7) and (3.8), with η denoting the learning rate of the stability network.

$$D(s_t, a_t) = c_{t+1} + \gamma d_z(s_{t+1}) - d_z(s_t) \tag{3.4}$$

$$A'(s_t, a_t) = A(s_t, a_t) - \omega D(s_t, a_t) \tag{3.5}$$

$$\Delta\theta = \alpha \nabla_\theta (ln\pi_\theta(s_t, a_t)) A'(s_t, a_t) \tag{3.6}$$

$$\Delta w = \beta \nabla_w (v_w(s_t)) A(s_t, a_t) \tag{3.7}$$

$$\Delta z = \eta \nabla_z (d_z(s_t)) D(s_t, a_t) \tag{3.8}$$

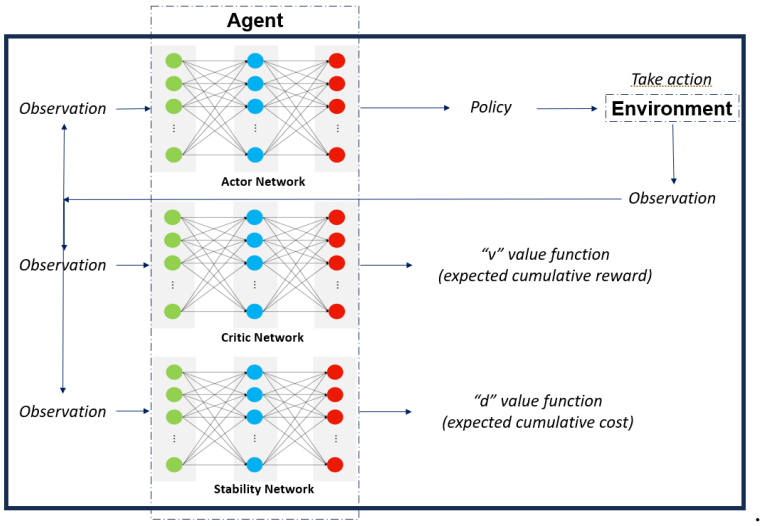

Fig. 3.8 The AWAC framework.

Specifically, ω is defined as a sigmoid function, $\sigma(\xi)$, where ξ is the intermediate variable between the agent's training progress and the ω value allocation. The definition of the designed sigmoid function is presented in (3.9).

In order to exploit the meaningful region of the sigmoid function and maintain the effectiveness in the ω adjustment, the aim is to stay outside the saturation areas of the sigmoid function. Therefore, a limit is applied to ξ, such as $-5 \leq \xi \leq 5$. The training process begins by allocating an initial value to the intermediate variable, $\xi = -5$, which leads to $\omega = 0.1$. When ω is close to 0.1 (initial value), rapid ω changes are avoided, since the agent is still exploring and it is allowed to interact more freely with the environment. At the same time, when ω is close to 0.5, rapid ω changes are also avoided, since the agent should persistently be discouraged to repeat poor policies. In other cases, the behaviour of the ω value changes is close to linear.

$$\omega = \sigma(\xi) = 0.1 + \frac{0.4}{1 + e^{-\xi}} \tag{3.9}$$

During training, the current mean expected cost, EC_{curr}, is estimated and compared against the mean expected cost of the previous training update, EC_{prev}, in order to monitor the direction of the training's stability. Then, an increment, $\Delta\xi_{incr}$, is applied if the cost is heading towards higher values, and a decrement, $\Delta\xi_{decr}$, otherwise, in the way presented in (3.10) and (3.11) respectively. This tuning of the ξ variable is always within the limitation of $-5 \leq \xi \leq 5$. The pseudo-code of the proposed algorithm is shown in Algorithm 1.

$$\Delta\xi_{incr} = +\frac{1}{2}\frac{EC_{curr}}{EC_{prev}} \tag{3.10}$$

$$\Delta\xi_{decr} = -\frac{1}{2}\frac{EC_{prev}}{EC_{curr}} \tag{3.11}$$

Algorithm 1 Adaptive Weighted Actor Critic

Require: timesteps N, batch size B, update epochs K
  Initialise neural network parameters θ, w and z
  Initialise learning rates α, β and η (annealing)
  Initialise stability parameters ξ and ω
  for $update = 1, update{+}{+}, update < N/B$ **do**
    for $t = 1, t{+}{+}, t < B$ **do**
      Sample $s_t, a_t\,(from\,\pi_\theta), s_{t+1}, r_{t+1}, c_{t+1}$
      Approximation: v_w, d_z
    end for
    for $t = B, t{-}{-}, t > 1$ **do**
      Calculate $A(s_t, a_t), D(s_t, a_t)$
    end for
    Calculate expected future returns
    if $update >= 2$ **then**
      Compare EC_{curr} and EC_{prev}
      $\xi \leftarrow \xi + \Delta\xi$
      $\omega \leftarrow \sigma(\xi)$
    end if
    Calculate $A'(s_t, a_t)$
    for $i = 1, i{+}{+}, i < K$ **do**
      Create random minibatches from batch
      for $each\,gradient\,step$ **do**
        $\theta \leftarrow \theta - \Delta\theta$
        $w \leftarrow w - \Delta w$
        $z \leftarrow z - \Delta z$
      end for
    end for
  end for

Intuitively, instead of using the AWAC method, one could attempt to manipulate the reward function accordingly, i.e. $r'_{t+1} = r_{t+1} + \omega\, c_{t+1}$. The reward function manipulation is a straightforward approach to provide a purely heuristic stability improvement, however, the proposed AWAC method has some significant benefits, which are discussed below:

- In AWAC, the ω factor is auto-tuned during training and it is constantly updated according to the training's progress. With reward manipulation, this tuning would have to be manually implemented, based on a series of necessary experiments beforehand that would determine an arguably suboptimal value for ω.
- In AWAC, the training of the critic network and the training of the stability network occur independently. The stability network only influences the policy update without complicating further the critic network's training. With reward manipulation, however, both the performance metric and the stability metric would be included in the reward function, hence they would be both included in the training of the critic's network.
- In AWAC, the reward graph is straightforward to interpret since the reward consists solely of the performance metric of the problem. With reward manipulation, the reward graph would become more

complex to understand, hence it would be challenging to investigate the contribution of each reward factor in the agent's performance.

A substantial benefit of AWAC is that its RL framework does not require any additional hyperparameters to be manually tuned. Thus, it is simple to implement and interpret its performance. The only intervention by the user is the construction of a stability cost function, which guides the agent towards its goal, assisting the originally constructed reward function within the RL framework. The construction of a stability cost function requires similar intuition to the one corresponding to the construction of any reward function, hence, there is no inherently added complexity in the proposed method. One could include this stability cost function in the originally constructed reward function of SAC and gain similar results with AWAC. However, this would raise a number of interpretation issues and tuning challenges. Hence, the benefit of AWAC is that it yields a more intuitive approach to introduce stability in the RL training than reward manipulation.

3.5 SLM process control results

3.5.1 Control problem

In this section, the benefits and the challenges of layer-wise control in SLM are demonstrated, via case studies of applications in symmetric parts. The control is implemented by varying the power, layer-by-layer, in order to achieve the desired average layer temperature of the meltpool. The geometry of the part plays crucial role regarding the level of challenge in this control task. If the layers of the part are long and wide enough, then the heat dissipates effectively and the heat accumulation observed among layers is small. However, when addressing thinwall structures (e.g., thin design features in heat exchangers), the heat accumulation observed among layers is significant, and changing the power according to a control law can be crucial for the quality of the produced part. Hence, in this demonstration the focus is on a thinwall geometry.

Specifically, the geometry of the part is a cuboid, with a base that consists of 4 tracks, with 10mm length each, resulting in a thinwall geometry. The cuboid consists of 35 layers. The simulation model is set up to take a value of power for each layer as an input, and produce for this layer a time series of thermal history, consisting of 800 points. This time series per layer is then averaged to be used as a performance indicator (and target for control). Each layer comprises a timestep from the controllers' perspective. When applying RL, in terms of RL notation and terminology, see [40], the whole build is referred to as an episode. Hence, it follows that the whole build is an episode consisting of 35 timesteps (35 layers). The RL implementation details for the following case studies are presented in Table 3.2.

Three different control approaches are compared; the state of the art SAC reinforcement learning, the stable approach AWAC reinforcement learning, and these are benchmarked against a PID controller. Regarding the RL techniques, SAC and AWAC, the agents undergo a training process before they come up with their resulting policy for the control problem. Hence, it is essential to compare the SAC and AWAC agents' training process and assess the impact of AWAC in the stability of the training. The improvement in stability is measured with regards to reduction in RL training variance, both overall and during pre-convergence training periods, while achieving higher or at least the same reward values. After the training is completed, the resulting policies of SAC and AWAC are compared, against the control policy of a carefully tuned PID controller. The training of SAC and AWAC is a stochastic process, hence the resulting policy of the agents

is different for each time the same experiment is run. For the PID, however, the resulting policy is always the same (deterministic) for the same experiment. Hence, in order to compare the SAC and AWAC training processes, multiple experiments are run and statistical analysis is performed. For comparison, the control policy for the PID is contrasted against the average resulting policy for the SAC and the AWAC agents (rather than the best resulting policy). The control policies are tested in two scenarios, in a target tracking setting, in which the target is fixed, as well as a setting in which the target varies with time. The hyperparameters used for the SAC and the AWAC agent are shown in Table 3.3.

Table 3.2 RL implementation summary

Case study	Action space Box(1,)	State space Box(3,)	Episode duration
Fixed target	Power[250, 300]	Temperature, Power, Height	35 timesteps (layers)
Tracking target	Power[250, 350]	Temperature, Power, Height	35 timesteps (layers)

Table 3.3 Hyperparameters for SAC and AWAC training

RL agents	Learning rates	Buffer/Batch size	Discount factor	Entropy coefficient/ω factor
SAC	3e-4	1e6	0.99	auto
AWAC	3e-4	2048	0.99	auto

3.5.2 Thinwall control with fixed control target

The first case study is process control for a thinwall geometry, with the aim of maintaining the average layer temperature constant throughout the build. T_{melt} is defined as the average layer temperature observed and A_{melt}, as the average meltpool area observed during the simulated build of a layer. The desired temperature is set to be T_{target}= 2308K and the SAC controller framework is designed as follows:

- Action space: The action space is a continuous space and it includes only the power of the laser beam, 250W$\leq P \leq$300W.
- State space: The state space is a continuous space. It consists of the observed average layer temperature, T_{melt}, the power, P, that was used to achieve this temperature, and the part's current height (layer).
- Reward function: The reward function plays a crucial role to RL training. Based on the work of [14], the reward function r, per layer, is formulated as:

$$r = 1 - \left| \frac{T_{target} - T_{melt}}{100} \right| \tag{3.12}$$

In order to implement the AWAC control approach, the same action space, state space and reward function are used as in the case of SAC. All the environment's definitions and assumptions remain the same. However, there is a need for a stability metric to indicate if the agent is far or close to the control target. The stability metric in this case study is chosen to be the meltpool area, since the meltpool area is correlated with the

meltpool temperature, which is the control target variable (one may select a different stability metric). The desired temperature is still set to be T_{target}= 2308K and the corresponding desired average meltpool area is calculated to be A_{target}= 5.44e−8m^2. For the cost function c, per layer, a quadratic Lyapunov term is constructed, as it is also common in control theory, see [78]. In this case, it is formulated as:

$$c = \left(\frac{A_{target} - A_{melt}}{0.01\text{e}{-8}} \right)^2 \tag{3.13}$$

10 separate experiments are run for each agent (SAC and AWAC). Figure 3.9 depicts the training process of the SAC and AWAC agent. The average reward and the standard deviation are shown by the dense line and the error bars respectively. It is observed that the agents achieve high levels of reward, since they reach a maximum higher than 31.5 out of 35 (maximum theoretically possible), hence higher than 90% training performance. The AWAC agent seems to outperform the SAC agent in stability and overall robustness of the training. More specifically, the comparison metrics and the results for the training are shown in Table 3.4. Similar to the step response settling time in control theory, the settling time here refers to the timestep in which the agent reaches 95% of the maximum reward and stays consistently above the 95% for the rest of the training (convergence). The mean settling time std refers to the mean of the standard deviations calculated from the beginning of the training until the settling time.

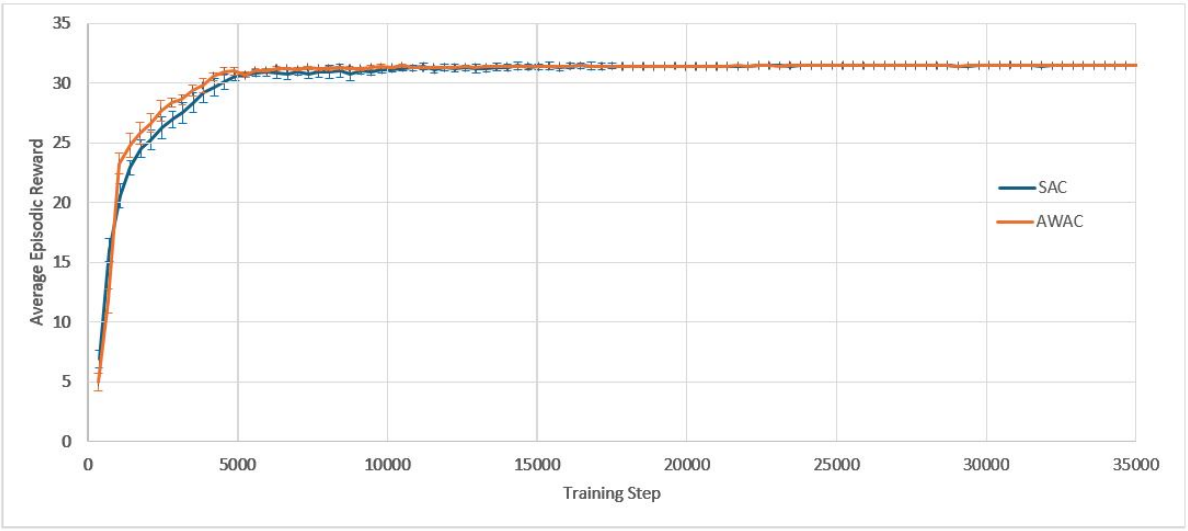

Fig. 3.9 Training curves of the SAC and the AWAC agents. Fixed target for the average layer temperature.

At this point, it is worth investigating the training process of SAC and AWAC more in depth, in order to gain intuition and fully understand the benefits of the AWAC approach. As observed in Table 3.4, the maximum rewards achieved in the two training processes show no significant difference between SAC and AWAC, while both training processes converge to the same reward values. Hence, it is expected that the resulting policies of the two approaches after the training completion show no significant difference. However, after the early, preliminary training steps (within which the interactions are mostly random for both agents), it is observed that AWAC reaches consistently higher rewards than SAC during the pre-convergence period. In a realistic scenario, in which computational cost and resources availability could be critical factors, one might not be able to train the RL agents until full convergence. Hence, it is plausible that the training process would

have to stop during the pre-convergence period. For this reason, an additional experiment is run, where in the 10 separate training processes conducted before for each agent, see Figure 3.9, the training is terminated before the settling time, and the resulting policy of the two agents is investigated. Specifically, the training is stopped at 33.3% convergence and at 66.6% convergence and the resulting policies and temperatures of SAC and AWAC are shown in Figures 3.10, 3.11, 3.12 and 3.13. In the 33.3% scenario, it is observed that the AWAC agent comes up with a better policy to maintain the average layer temperature at the desired value, reaching a mean of 2306.9K, with a mean absolute error of 1.87K. The SAC agent also reaches satisfactory temperatures, with a mean of 2306.2K, however, the mean absolute error is 2.57K, which is approximately 38% worse than AWAC. In the 66.6% scenario, it is observed once more that the AWAC agent comes up with a better policy, reaching a mean of 2307.9K, with a mean absolute error of 1.22K. The SAC agent also reaches satisfactory temperatures, with a mean of 2307.2K, however, the mean absolute error is 1.74K, which is approximately 43% worse than AWAC. In conclusion, the fact that AWAC reaches consistently higher rewards than SAC, in a more stable manner, has significant impact in the resulting policy of the two agents, and this impact is particularly shown in cases in which there are limitations in computational resources and cost.

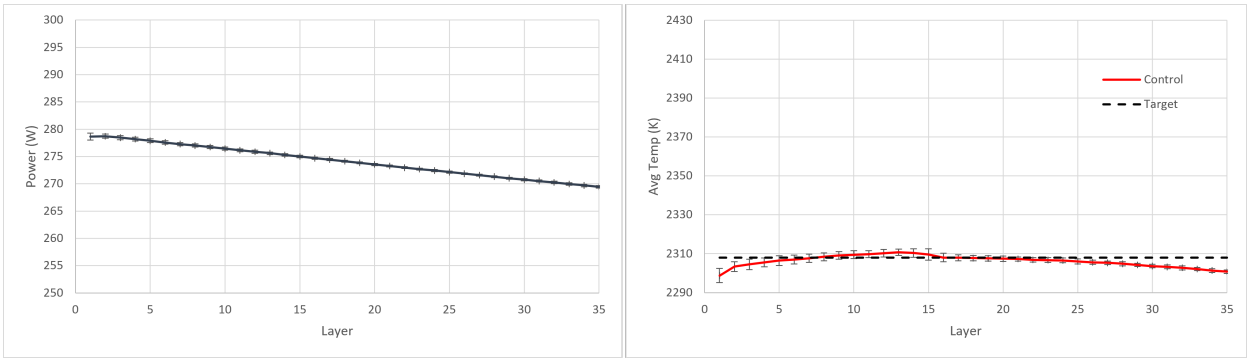

Fig. 3.10 Resulting policy of the SAC agent with fixed target. Training stopped at 33.3% convergence timesteps.

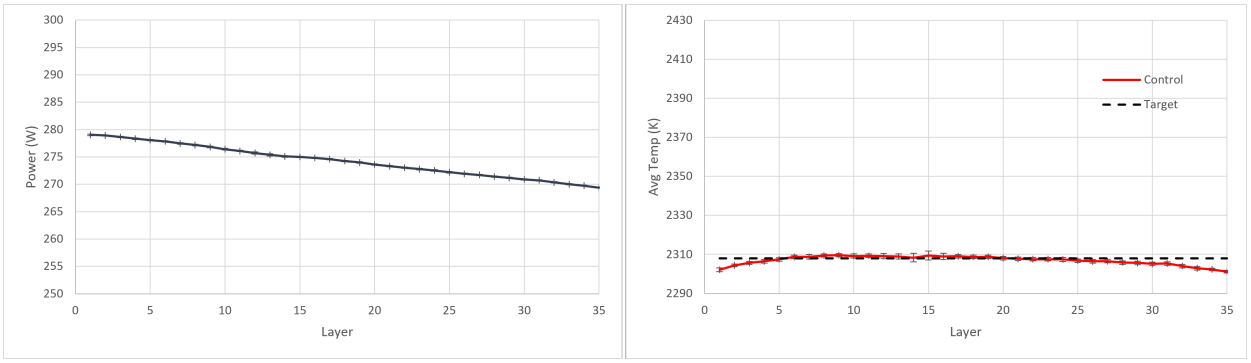

Fig. 3.11 Resulting policy of the AWAC agent with fixed target. Training stopped at 33.3% convergence timesteps.

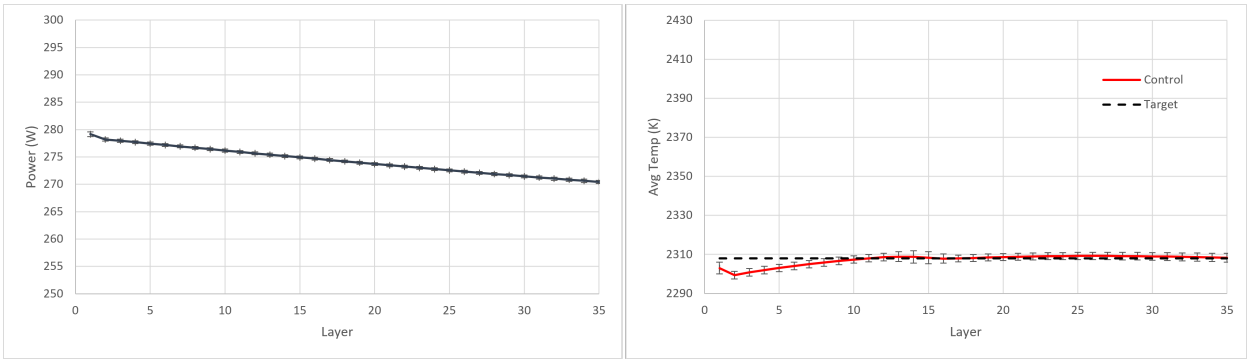

Fig. 3.12 Resulting policy of the SAC agent with fixed target. Training stopped at 66.6% convergence timesteps.

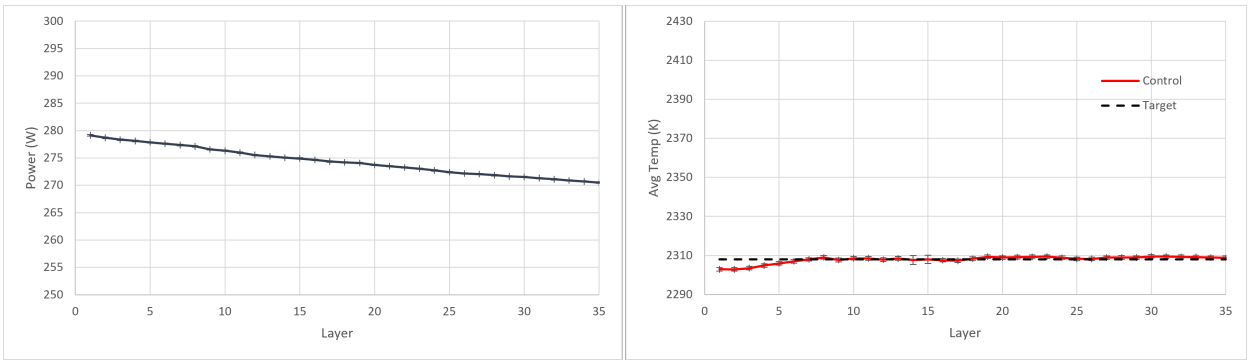

Fig. 3.13 Resulting policy of the AWAC agent with fixed target. Training stopped at 66.6% convergence timesteps.

Table 3.4 Fixed target training comparison between SAC and AWAC. All values correspond to the average of 10 runs

RL agents	Mean Reward	Mean of Std	Max Reward	Settling Time	Mean Settling Time Std
SAC	30.37	0.24	31.54	4550	0.80
AWAC	30.55	0.15	31.54	4200	0.70

In order to implement PID control, parametric optimisation via gradient descent is used for tuning the P, I, and D terms. These terms are then further fine-tuned heuristically which results in a further performance increase. The final values of the control law terms are $K_P = 5.58e-10$, $K_I = 1.01e-13$ and $K_D = -1.9e+01$. The resulting policy (from RL completed training and PID tuning) and the achieved temperature in each layer are shown in Figures 3.14, 3.15 and 3.16. It is argued that both RL controllers manage to maintain the average layer temperature in the desired value in a satisfactory manner throughout the build, outperforming the PID controller. It is noticed that the PID controller follows a reasonable policy of gradually decreasing power. However, it does not reach the desired value as effectively and consistently as the RL controllers, demonstrating a slight offset from the desired value. The offset challenge is a well-known challenge in PID controllers, see [79]. The full list of comparison metrics and the results for the controllers are shown in Table 3.5.

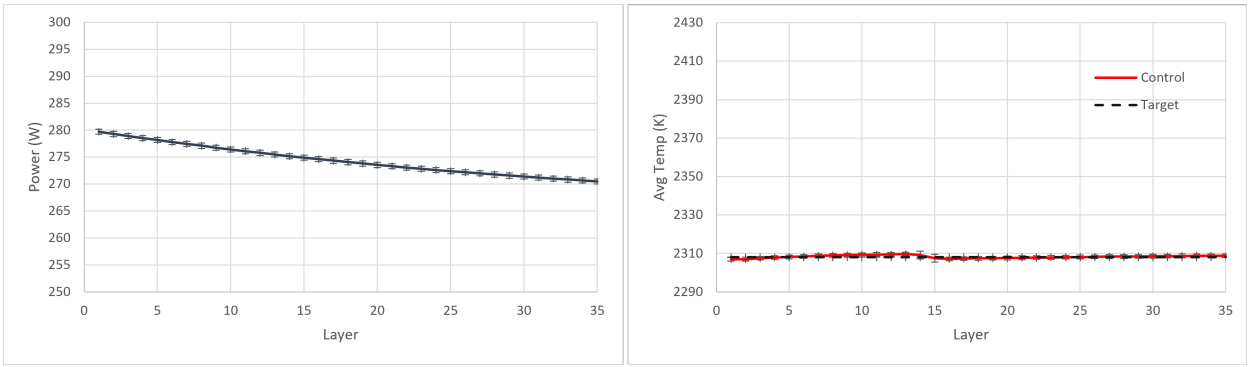

Fig. 3.14 Resulting policy of the SAC agent with fixed target.

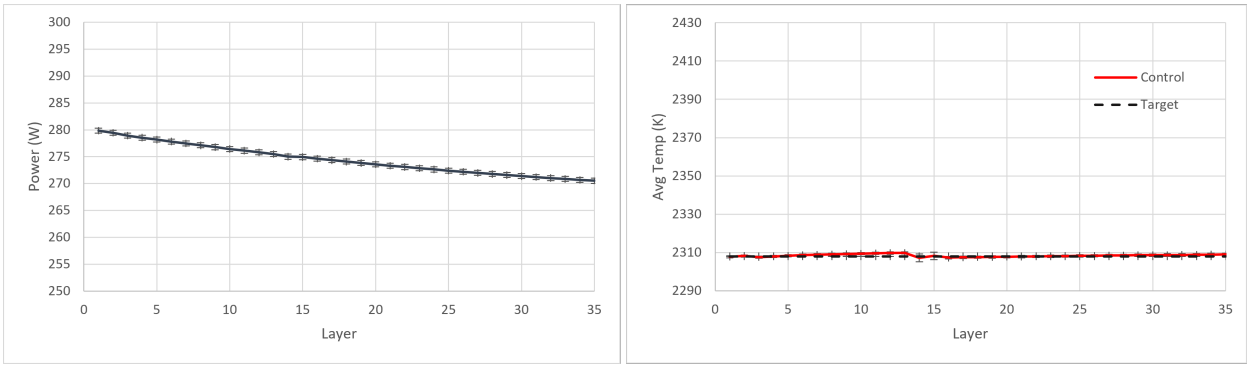

Fig. 3.15 Resulting policy of the AWAC agent with fixed target.

Table 3.5 Fixed target results comparison of PID, SAC and AWAC. SAC and AWAC values correspond to the average of 10 runs

Controller	Mean Temperature (K)	Mean Temperature Error (K)
SAC	2308.27	0.65
AWAC	2308.39	0.62
PID	2311.24	3.24

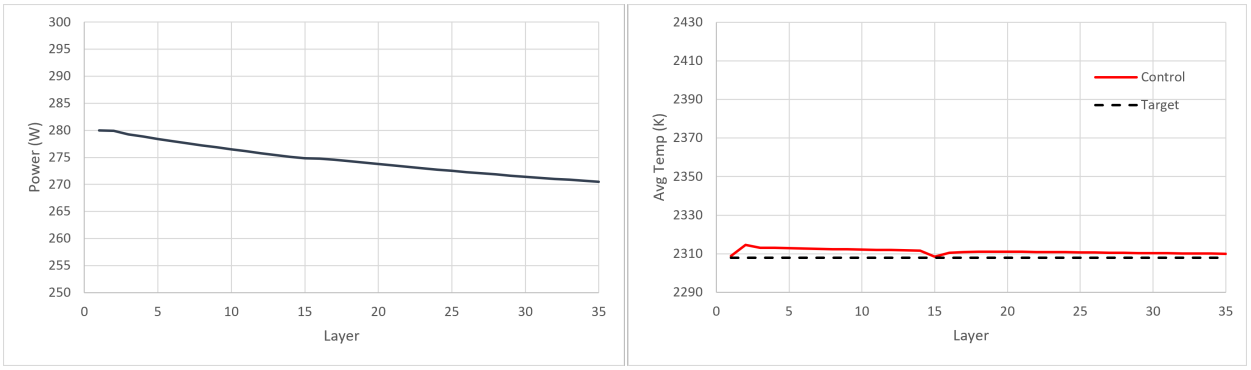

Fig. 3.16 Resulting policy of the PID with fixed target.

3.5.3 Thinwall control with tracking control target

The second case study attempts to challenge the controller, in the sense that the control target is no longer fixed. This could be the case, for example, when one wishes to create bespoke material microstructure and local properties via manipulating the cooling rate between layers. The desired temperature, T_{target}, now varies with the part's height, as seen in the results in Figure 3.18. Hence, the way the RL frameworks are defined is altered, as follows:

- Action space: The action space is a continuous space and it includes only the power of the laser beam, 250W$\leq P \leq$350W.
- State space: The state space is a continuous space. It consists of the observed average layer temperature, T_{melt}, the power, P, that was used to achieve this temperature, and the part's current height (layer).
- Reward function: The reward function plays a crucial role to RL training. Based on the work of [14], the reward function r, per layer, is formulated as:

$$r = 1 - \left| \frac{T_{target} - T_{melt}}{100} \right| \tag{3.14}$$

As the control target is no longer fixed, the stability metric target for AWAC needs to be redefined accordingly. Specifically, a step change to the meltpool area target is introduced that corresponds to each new meltpool temperature target, as $A_{target} = 5.44\text{e}{-8}\text{m}^2$, $A_{target} = 6.06\text{e}{-8}\text{m}^2$, $A_{target} = 5.10\text{e}{-8}\text{m}^2$, $A_{target} = 5.91\text{e}{-8}\text{m}^2$, $A_{target} = 6.40\text{e}{-8}\text{m}^2$, $A_{target} = 5.26\text{e}{-8}\text{m}^2$, and $A_{target} = 5.44\text{e}{-8}\text{m}^2$. For the cost function c, per layer, a quadratic Lyapunov term is constructed, as it is also common in control theory, see [78]. In this case, it is formulated as:

$$c = \left(\frac{A_{target} - A_{melt}}{0.01\text{e}{-8}} \right)^2 \tag{3.15}$$

10 separate experiments are run for each agent (SAC and AWAC). Figure 3.17 depicts the training process of the SAC and AWAC agent. The average reward and the standard deviation are shown by the dense line and the error bars respectively. It is observed that the agents achieve high levels of reward, since they reach a maximum higher than 30.6 out of 35 (maximum theoretically possible), hence higher than 87% training performance. The AWAC agent seems to outperform the SAC agent in stability and overall robustness of the

training. More specifically, the comparison metrics and the results for the training are shown in Table 3.6. Similar to the step response settling time in control theory, the settling time here refers to the timestep in which the agent reaches 95% of the maximum reward and stays consistently above the 95% for the rest of the training (convergence). The mean settling time std refers to the mean of the standard deviations calculated from the beginning of the training until the settling time.

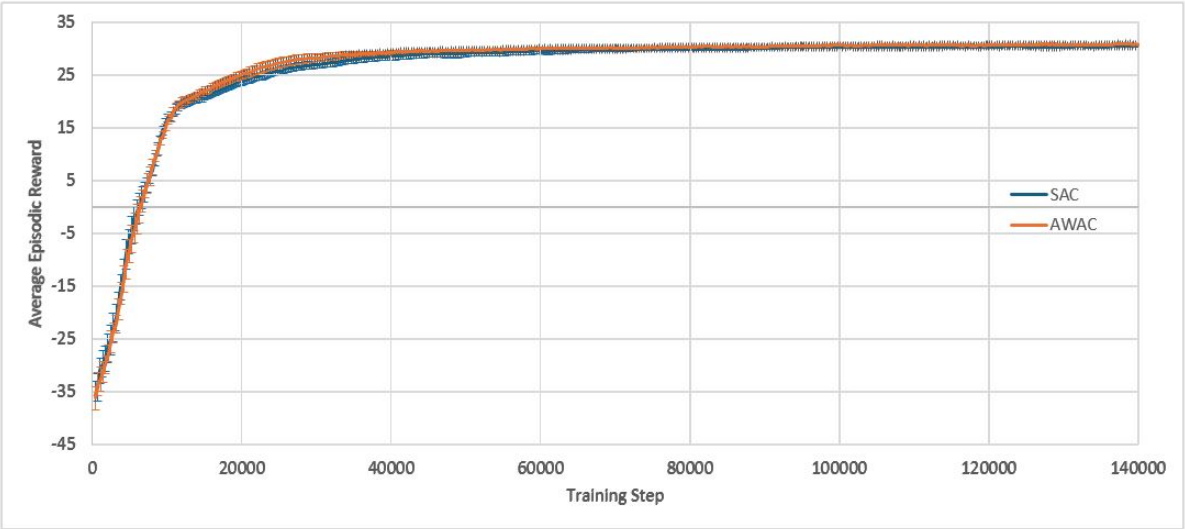

Fig. 3.17 Training curves of the SAC and the AWAC agents. Tracking target for the average layer temperature.

Table 3.6 Tracking target training comparison between SAC and AWAC. All values correspond to the average of 10 runs

RL agents	Mean Reward	Mean of Std	Max Reward	Settling Time	Mean Settling Time Std
SAC	26.28	0.54	30.65	44800	0.98
AWAC	26.51	0.35	30.83	40950	0.94

For the PID controller, the same formulation and tuning are used as in the earlier case study, in section 4.2. The resulting policy and the achieved temperature in each layer are shown in Figures 3.18, 3.19 and 3.20. It is argued that both RL controllers manage to maintain the average layer temperature in the desired value in a satisfactory manner throughout the build, outperforming the PID controller. The PID controller demonstrates a delay in the action it takes when there is a change in the control target. This delay is expected (and a known drawback in PID control methods), see [79], since the controller does not know about the change in target, until this is fed back indirectly via the error signal. In contrast, the RL controllers demonstrate no delay and they follow the target effectively. More specifically, the comparison metrics and the results for the controllers are shown in Table 3.7.

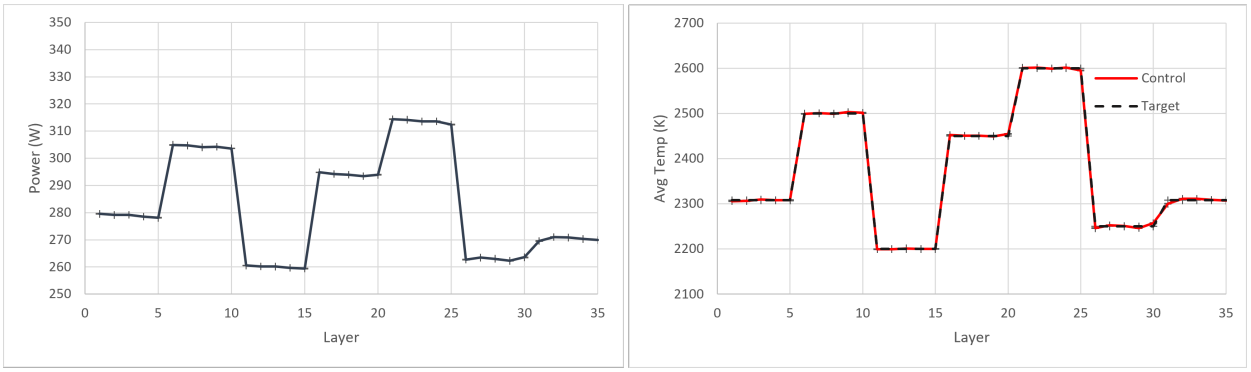

Fig. 3.18 Resulting policy of the SAC agent with tracking target.

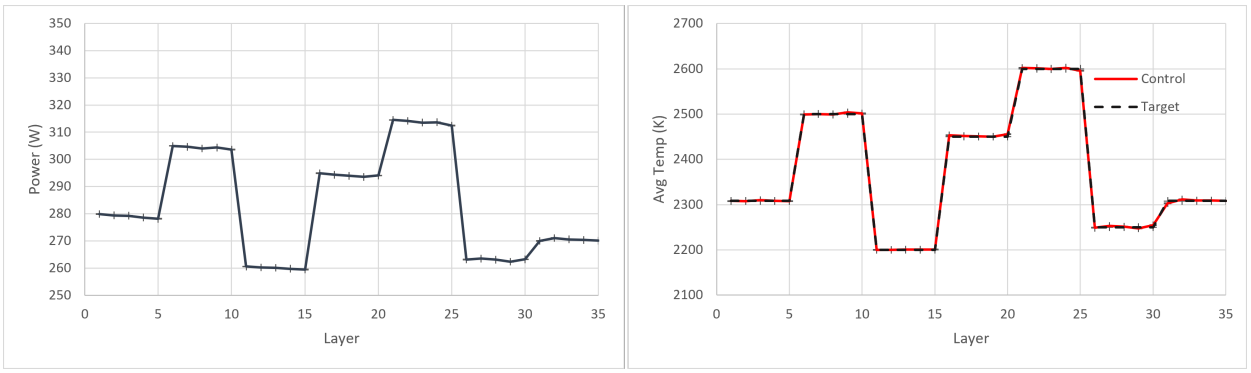

Fig. 3.19 Resulting policy of the AWAC agent with tracking target.

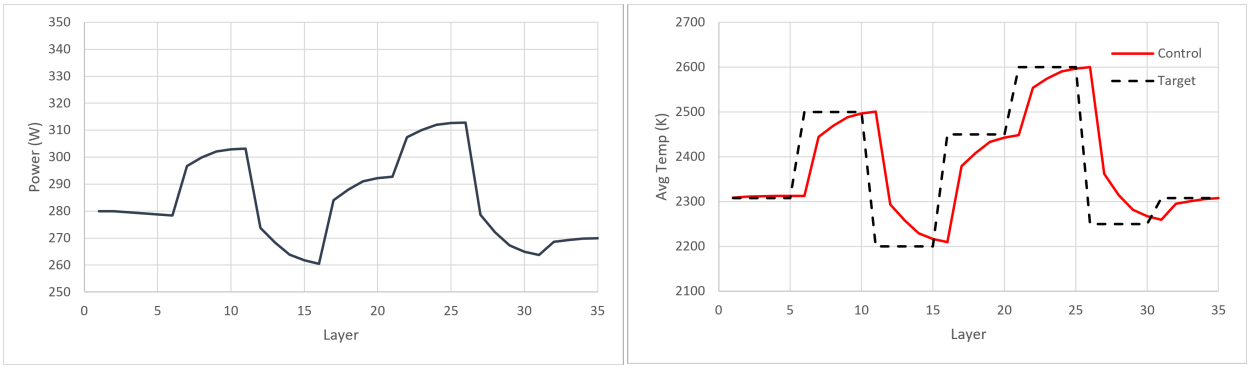

Fig. 3.20 Resulting policy of the PID with tracking target.

Table 3.7 Tracking target results comparison of PID, SAC and AWAC. SAC and AWAC values correspond to the average of 10 runs

Controller	Action delay	Mean Temperature Error (K)
SAC	No	1.95
AWAC	No	1.63
PID	Yes	59.05

3.6 Discussion and future research directions

For the first time in the literature, in this study we demonstrate the benefits of a RL process control framework for multiple layers (complete 3D PBF parts) and we highlight the importance of stability during training. The presented case studies confirm the effectiveness of the proposed control framework, directly addressing heat accumulation issues while demonstrating effective overall process control. Based on simple 3D part geometries, SAC and AWAC outperform a tuned PID controller. Moreover, when comparing the RL algorithms' training, we confirm the benefits of the proposed AWAC approach regarding stability and consistent performance, which can be key factors for satisfactory manufacturing results and practical applications.

Despite the satisfactory control performance of RL in our case studies, there are still important challenges that prevent RL from entering the control mainstream in the metal AM industry. As our case studies show, the RL agent needs to be trained before it derives a control policy. However, there is no established way to predict if the training is going to be successful. Further research, new methods and new analyses are needed in the areas of convergence guarantees, constraint satisfaction, and control performance guarantees, to create more trustworthy implementations of RL (which are required for critical applications).

From a process perspective (PBF), a noteworthy challenge is the development of advanced control strategies, for example to create bespoke microstructure and localised part properties. While this work focuses on a layer-by-layer control approach, there is a need to explore track-by-track and point-by-point control frameworks. However, these would require more complex models, that would have higher computational demands. We also expect that the identification of good control policies would be more challenging, hence this would require more effective and efficient RL methods.

Finally, in this work, we formulate the meltpool behaviour in a deterministic fashion. However, in a real-world PBF process, the meltpool behaviour is not deterministic, since there is uncertainty within the manufacturing process and sensitivity limitations in the monitoring capability and systems. In order to make the PBF simulation more realistic and test the controllers' capabilities, there is a need for testing against monitoring noise and other process disturbances. Hence, further research is required towards validation of simulation results with real-world experiments, as well as practical implementation for a range of part geometries and materials. This would also lead to further research work in RL, towards methods that are more tolerant to uncertainty and disturbances.

3.7 Declarations

3.7.1 Funding

This study was funded by the UK Engineering and Physical Sciences Research Council (EP/P006566/1) and iCASE (EP/T517835/1), with contributions from Wayland Additive Ltd.

3.7.2 Acknowledgements

The author Taha Al-Saadi would like to thank J. Anthony Rossiter (Department of Automatic Control & Systems Engineering, University of Sheffield, United Kingdom) for discussions and feedback that helped further improve the research work of this manuscript.

3.8 Compliance with ethical standards

The authors have no further ethical issues to disclose.

Chapter 4

Constrained reinforcement learning for advanced control in powder bed fusion

In the two previous chapters, a thorough investigation in RL stability is presented, along with a novel RL framework to enhance stability and RL performance. However, even in such stable RL frameworks, there is still no established way to predict how the RL agent is going to reach satisfactory performance, i.e. what actions (policy) the RL agent is going to develop during training and during control deployment towards its path to reach the desired control target. The actions that the agent attempts might include unreasonable patterns, or worse, they might not even be feasible in a given application. Hence, the result of such patterns can vary between simply unsatisfactory performance, and more critical consequences such as stress-testing/breaking a machine's physical limitations.

For this reason, a RL framework including constraint handling is necessary in any given critical application. In such a RL framework, the goal is the achievement of the desired control target while satisfying the imposed constraints. Especially in critical applications, such as PBF, a zero constraint violation framework is necessary, either not to compromise the integrity of the built part, or more importantly to respect the machine's limits and guarantee safety during the manufacturing process.

This chapter provides a thorough investigation in current constraint handling attempts in RL process control. After reviewing the current state of the art, a novel constrained RL algorithm is introduced, particularly tailored for the PBF process needs. In this particular RL framework, a squashing method is utilised in order to constrain the actions of the agent to the desired boundaries. Moreover, a scalar value is included in the squashing rule, as shown in this chapter, section 4.4.2, in order to tune the intensity of the squashing depending on the application's needs. Finally, the proposed algorithm is tested on a simulated PBF platform, in order to verify the zero constraint violation claim, and highlight the importance of the squashing tuning.

Note: the presented paper is slightly amended for the purposes of this thesis. The original paper has been peer reviewed and accepted in the European Control Congress (ECC), 2025: Stylianos Vagenas and George Panoutsos. Constrained reinforcement learning for advanced control in powder bed fusion. In 2025 European Control Conference (ECC), pages 1828–1835, 2025.

https://doi.org/10.23919/ECC65951.2025.11186875

4.1 Abstract

Reinforcement Learning (RL) continues to attract considerable attention in academia and industry. Its data-driven nature, combined with its varied and flexible formulation, makes it applicable in a variety of complex control tasks, in which control theory techniques can be challenging to implement. The Powder Bed Fusion (PBF) process, comprises an example of such a complex control task. However, there are still critical challenges in RL that need to be addressed in order to fully enable its use in PBF implementations. For instance, while constraint satisfaction comprises a necessity in PBF process control, there are still gaps in demonstration and analysis of RL algorithmic behaviour and control performance under constraint satisfaction. Existing constraint techniques in the literature, such as radial squashing, can provide zero constraint violation guarantees in process control. However, a constraint framework that also accounts for satisfactory control performance must be established. In this work, we attempt to address the above challenges, providing a thorough analysis on constrained RL for PBF process control, and assessing the impact of the radial squashing technique. The results of our analysis show that tuning the intensity of radial squashing can be vital for maintaining satisfactory control performance under constraints.

4.2 Introduction

Reinforcement Learning (RL) is a trial-and-error optimisation technique, based on mapping states to actions [40]. The RL controller, or agent, functions as a dynamic decision maker, constantly learning and improving its behaviour, and attempts to understand and control the environment via meaningful interactions with it [40]. This occurs in an iterative fashion, which leads to a trial-and-error feedback loop, driven by the goal of maximising a reward signal [80]. The most popular methods in RL process control belong to the actor-critic class of methods, with Soft Actor Critic (SAC) [81] standing out as state of the art algorithm. The efficiency and the dominance of such methods is demonstrated for a variety of standard benchmark tasks in the works of [41], [42] and [43].

Intricate continuous processes, such as Powder Bed Fusion (PBF), differentiate strongly from the aforementioned benchmark tasks, due to their inherently added complexity and dynamic behaviour [82]. PBF comprises an original, complex manufacturing technique for making 3D parts from metallic powder. The parts are made on a layer upon layer fashion, with a power source selectively melting the metallic powder, based on computer-designed models [69].

Within PBF, process control can be vital and it requires special attention to enable the process to realise its potential and achieve unique microstructures, see [83] and [84]. As the required parts become more intricate, advanced control methods are needed that often require integrated models and strict assumptions. In PBF, such models (integrated, across scales) do not exist [71], or certain signal property assumptions cannot be satisfied. Using RL to control the process is an alternative to control theory techniques, since RL offers a straightforward control framework, for which model development and assumptions can be more flexible, see [71] and [85]. Although RL does not provide the theoretical performance guarantees that control theory techniques provide, its flexible formulation has unlocked the investigation for RL process control in PBF. For instance, [71] and [14] demonstrated RL process control in a PBF simulation study, while [85] utilised

the SAC algorithm for process control in PBF, and they also introduced a new, more stable RL framework. Finally, [39] recently discussed the potential of RL algorithms in industrial control.

The above evidence suggests that there is a strong, growing interest in flexible RL frameworks for effective process control in industrial tasks, and particularly in PBF. However, there are still major challenges that need to be addressed. For instance, constraint handling comprises a necessity in PBF processes, while, as suggested by [39], constraint handling remains a substantial challenge in RL control. Examples of such constraints in PBF are dynamic action constraints that refer to microstructure issues, e.g., avoidance of steep power changes that would violate the part's integrity, or machine capabilities, e.g., incapability of steep power changes due to PBF machine limitations.

In RL, defining overall action (e.g., power) limits is almost ubiquitous in continuous control as found in all benchmark control tasks [86]. Existing implementations of RL commonly handle such limits using a hyperbolic tangent function as final activation layer of policies, which is referred to as squashing [53]. However, this simple squashing implementation does not account for dynamic, in-process constraints. Specifically, dynamic constraints are currently not accounted for in any state of the art RL framework, hence, there is no generally accepted method of guaranteeing such action constraints.

In this work, accounting for the aforementioned need for smooth power transitions in PBF, the focus is on action-based constraints, i.e. action constraints that depend on the previous action taken. A thorough literature review is conducted to investigate existing, in-process, constraint handling methods for RL control. Then, a new action-based constrained RL framework for PBF process control is developed, utilising a more intuitive radial squashing approach, in which the intensity of the radial squashing can be tuned. Finally, multiple experiments are conducted on a PBF simulation platform, under different constraints, assessing the impact of the radial squashing tuning on control performance.

4.3 Background

As previously mentioned, constraint handling comprises a substantial challenge in RL process control. The work of [51] was the first to introduce the constraint handling challenge in RL, by presenting the Constrained Markov Decision Process (CMDP). It was argued that, intuitively, the CMDP can be solved using the standard Lagrangian method [87]. However, [53] and [52] argued that while the Lagrangian method is asymptotically safe, the constraints are prone to be violated during RL training. Thus, research started to direct towards the Lyapunov methods from control theory. More specifically, [52] developed a Lyapunov barrier function to restrict the policy update to a safe set for each RL training iteration. Their method was also integrated in already existing state of the art algorithms, such as proximal policy optimisation, and it significantly reduced the number of constraint violations during training when compared to other state of the art baselines. [65] provided a Lyapunov, state trajectory based CMDP solution, which reached near-zero constraint violation. Building on this work, [88] provided a more straightforward Lyapunov, state-based CMDP solution. Finally, [53] discussed the potential of projection methods (safety layers), such as radial squashing, in order to provide zero constraint violation during RL training and control performance. For a more comprehensive literature review on RL constraint handling, the works of [39] and [89] are recommended.

After investigating the aforementioned literature, it is concluded that positive steps have been made towards addressing the challenge of constraint handling in RL. However, it is noted that most of the action

constraint attempts are state-based, i.e. either based on a full state trajectory or adjusted according to an absolute state by state sequence, whereas action-based constraints are in short in the RL literature. Moreover, despite comprising an improvement to the state of the art baselines, most of the attempts in the literature are unable to provide a zero constraint violation framework. Particularly in the Lyapunov methods, the RL agent needs to interact with the environment and collect enough interaction samples for a sufficient state-action mapping, which dictates what the safe policies/actions are. However, these preliminary, but necessary, interactions are inherently random. Hence, there is no guarantee of zero constraint violation during the sample collection. Moreover, in some cases, the satisfaction of constraints depends on how accurately this preliminary state-action mapping can be generalised (accuracy of regression). Hence, these methods, while generally effective, are inherently prone to constraint violation, particularly during RL training and sample collection.

In PBF, it is crucial to maintain a zero constraint violation framework, since any potential constraint violation can lead to damaged parts. The projection/squashing methods, although not comprising the most elegant solution, are the only methods found in the literature within which zero constraint violation can be guaranteed. Specifically, the radial squashing method can be combined with state of the art RL algorithms [53], such as SAC, while it is not prone to RL implementation challenges (e.g., vanishing gradients). As a result, the radial squashing method is chosen for constraint handling in this work.

4.4 Methodology

4.4.1 The reinforcement learning control framework

In the RL framework, the process control problem can be defined as a Markov Decision Process (MDP), which, as described in [40] and [71], is represented by the following tuple (S, A, r, Pr, γ), where:

- S is the state space that characterises the environment.
- A is the action space available to the agent.
- r is the reward provided by the environment when the agent takes an action $a \in A$ in state $s \in S$.
- Pr is the dynamics function $Pr(s', r|s, a)$, which gives the probability of transitioning to next state s' and achieving reward r, given the agent takes action a in state s. The timestep is represented as t, thus the next state can be written as $s' = s_{t+1}$, the reward as $r = r_{t+1}$, the current state as $s = s_t$ and the action as $a = a_t$.
- γ is the discount factor, which is a scalar value between 0 and 1 $(0 \leq \gamma \leq 1)$ and determines the weight of future rewards.

The cumulative discounted reward, G, can now be defined as $G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ...$, and the action-value function, $q_\pi(s,a)$, can be defined as the expectation, E, of cumulative discounted reward following a policy, π, i.e. $q_\pi(s,a) = E_\pi[r_{t+1} + \gamma G_{t+1}|s_t = s, a_t = a]$. The policy π denotes the state-action mapping of the agent. The agent's goal is to to maximise the above expectation, finding the optimal action-value function $q_*(s,a) = max_\pi q_\pi(s,a)$.

Following the example of [85] for RL control in PBF, the SAC algorithm is chosen for this work, as introduced by [42]. SAC is a state of the art RL algorithm, which aims not only to maximise the aforementioned reward expectation but also maximise entropy and enhance the exploration of control actions. It consists of an actor neural network responsible for policy updates, determining action selection, and two critic neural networks responsible for evaluating the quality of the taken actions.

4.4.2 The radial squashing method

The novelty of the proposed methodology in this work, is within the way the constraints and the radial squashing rules are formulated. Specifically, radial squashing has been utilised for state-based constraints in the literature, whereas in this work, it is utilised for action-based constraints, and the feasible action region is determined by the previous action taken (smooth power transitions in PBF). Moreover, radial squashing is combined with the SAC algorithm, taking the role of an extra neural network layer (constrained RL framework), as presented in Figure 4.1. Finally, in this work, for the first time in the literature, a scalar, positive hyperparameter, K, is introduced in the squashing rule, in order to adjust the intensity of the squashing (radial squashing tuning).

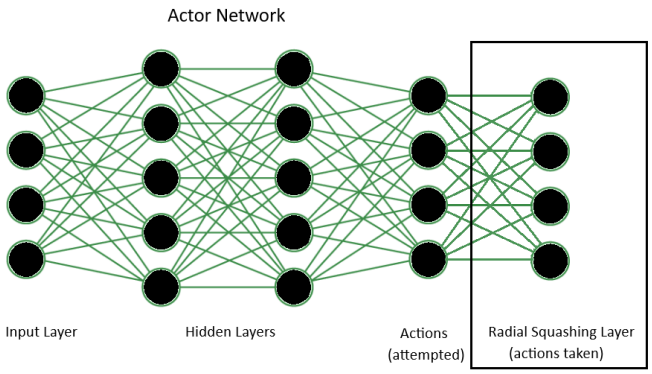

Fig. 4.1 Radial squashing as an output layer in SAC actor network.

More specifically, this radial squashing variant is described as follows. Let a_t be the taken action on timestep t. Based on the taken action a_t, the boundaries b_1 and b_2 are defined as constraint for the next action, with $b_1 < a_{t+1} < b_2$. If the next attempted action by the agent is $a'_{t+1} < a_t$, then the update rule in (4.1) is applied to satisfy the constraint, whereas if $a'_{t+1} > a_t$, then the update rule in (4.2) is applied. As a result, this attempted action is squashed into the feasible region, $[b_1, b_2]$, and the actual taken action, a_{t+1}, satisfies the constraint. Following this reasoning, the action a_{t+1} determines the new boundaries for the next action, a_{t+2}. Hence, every attempted action is mapped into its feasible region, assuring the desired smoothness of transitions from action to action and guaranteeing zero constraint violation on every timestep of the action trajectory. Figure 4.2 provides an illustration of the radial squashing method for a one-dimensional action space task. Without loss of generality, this method can be also used for a multi-dimensional action space task.

$$a_{t+1} = a_t + tanh\left(K\frac{|a'_{t+1} - a_t|}{|b_1 - a_t|}\right)(b_1 - a_t) \tag{4.1}$$

$$a_{t+1} = a_t + tanh\left(K\frac{|a'_{t+1} - a_t|}{|b_2 - a_t|}\right)(b_2 - a_t) \tag{4.2}$$

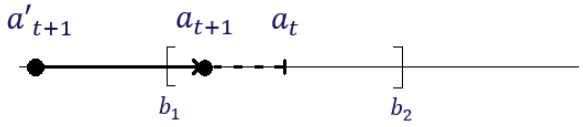

Fig. 4.2 Illustration of the action-based radial squashing method for a one-dimensional action space, as achieved by the actor network final layer.

The exact value of the next taken action depends on the value of the introduced K parameter. For instance, if the value of K is *low*, the next taken action approaches saturation to the center of its feasible region (i.e. the value of the previously taken action), whereas if the value of K is *high*, the next taken action approaches saturation to the boundaries of its feasible region (i.e. b_1 or b_2). Finally, for certain K values, a *good mapping* is allowed, in which the attempted actions are mapped into the feasible actions, while meaningfully exploiting the feasible region dictated by the boundaries.

Finding a suitable value for the K parameter is the main challenge of the radial squashing method, since the interpretation of *low* K and *high* K values is task-dependent. Thus, one has to develop an algorithm and define the necessary metrics, which are to determine the aforementioned distinctive K values for the specific task at hand.

A metric that shows the distribution of each taken action is the Absolute Differences Percentage (ADP), as presented in (4.3). The correspondent metric for the attempted action, is defined to be the ADP', as presented in (4.4). The d_{max} and d'_{max} values are the respective maximum, theoretically possible, absolute differences. Averaging the ADP and ADP' metrics, across all actions, gives two metrics, Av and Av' respectively, that describe how all the taken actions and all the attempted actions are distributed within their correspondent feasible regions. These two metrics can be used in order to determine the effective K region [*low*, *high*] and the K value that provides *good mapping* for the specific task. The whole procedure is described in detail, in Algorithm 2, while an example is also given in the following subsection for more intuition.

$$ADP_{t+1} = \frac{|a_{t+1} - a_t|}{d_{max}} \tag{4.3}$$

$$ADP'_{t+1} = \frac{|a'_{t+1} - a_t|}{d'_{max}} \tag{4.4}$$

Algorithm 2 Effective K region calculation

Require: number of trajectories N, number of timesteps T, positive step ΔK, d_{max}, d'_{max}
Generate N action a' trajectories of T timesteps each
Set desired boundary constraint
Initialise $K \leftarrow 0$
done←False
while not done **do**
  for $i = 1, i{+}{+}, i <= N$ **do**
    for $j = 1, j{+}{+}, j < T$ **do**
      if $j = 1$ **then**
        Set $a_{i,j} \leftarrow a'_{i,j}$
      end if
      Calculate $a_{i,j+1}$ from (1) or (2)
    end for
  end for
  for $i = 1, i{+}{+}, i <= N$ **do**
    for $j = 1, j{+}{+}, j < T$ **do**
      Calculate $ADP_{i,j+1}$ and $ADP'_{i,j+1}$
      from (3) and (4)
    end for
    $Av_i \leftarrow mean_j(ADP)$
    $Av'_i \leftarrow mean_j(ADP')$
  end for
  $low\,K \leftarrow last\,K\,with\,mean_i(Av) < 1\%$
  $good\,map\,K \leftarrow K\,with\,min[mean_i(|Av - Av'|)]$
  $high\,K \leftarrow first\,K\,with\,mean_i(Av) > 99\%$
  $K \leftarrow K + \Delta K$
  If $high\,K$ is found set done←True
end while

4.4.3 Powder bed fusion example

In this example, a PBF process is assumed to be the task at hand, defined as a one-dimensional action space task. The action is defined as the power source, P, with the overall limits of 200W$\leq P \leq$300W, which are typical values for power limits in PBF [18]. For simplicity, a boundary constraint of constant nature, $\pm$0.2W, is examined. For this boundary constraint, different K values are investigated, according to Algorithm 2. The results of Algorithm 2, applied for one random action trajectory, with $d_{max} = 0.2$, and $d'_{max} = 100$, are shown in Figure 4.3.

It is observed that the K parameter plays crucial role on the efficiency of the mapping. For instance, the value of $K = 0.0022$ provides *good mapping* when the boundary constraint is $\pm$0.2W, since the attempted actions are meaningfully mapped into the feasible region. However, the value of $K = 0.0001$ is *low* and the value of $K = 0.1031$ is *high* for the $\pm$0.2W constraint, giving an effective K region [0.0001, 0.1031]. By increasing the number of sampled random trajectories and decreasing the value of positive step ΔK, Algorithm 2 becomes more reliable and accurate, in the expense of higher computational cost.

By implementing Algorithm 2, this time for various boundary constraints, the behaviour of the Av metric (the distribution of the taken actions) can be examined with relation to the K parameter. The results of

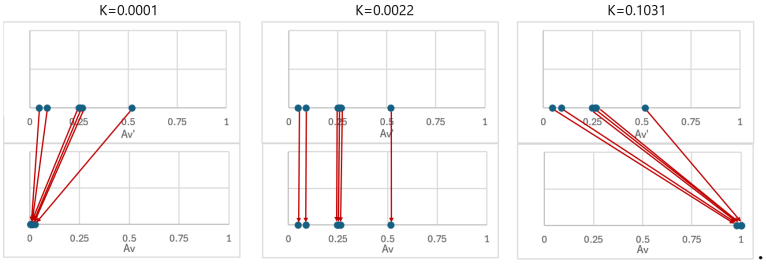

Fig. 4.3 Mapping efficiency of the radial squashing method for the boundary constraint of ±0.2W, at different K values, determined using Algorithm 2.

this implementation are shown in Figure 4.4 and it is observed that Av is related monotonically with the K parameter, regardless of the value of the boundary constraint. Moreover, it is noted that the larger the boundaries of the constraint, the larger the limit values of the effective K region [*low*, *high*]. For reference, for the same action trajectory, the ±0.2W constraint, gives an effective K region [0.0001, 0.1031], while the ±20W constraint, gives an effective K region [0.0091, 6.9602].

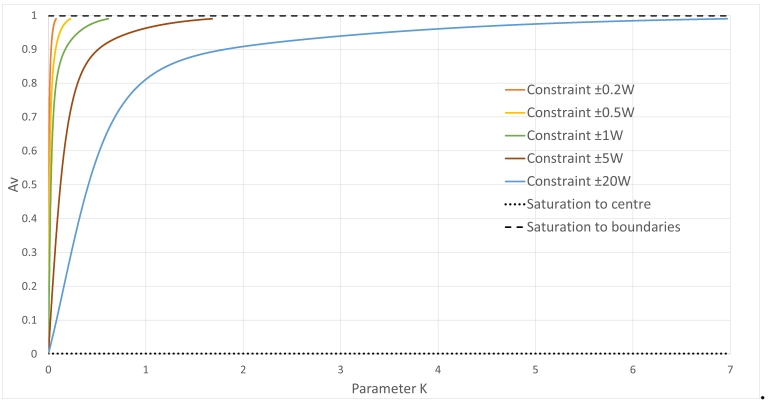

Fig. 4.4 Relation between Av and K parameter, under different boundary constraints, determined using Algorithm 2.

4.4.4 Radial squashing effect on process control

In certain constrained RL control tasks, instead of actions well mapped into the feasible region, actions saturated towards the center (*low* K) or towards the boundaries (*high* K) can be preferred by the agent in order to achieve the best control performance. Thus, granted the effective K region, an investigation within this region needs to be conducted in order to determine the K values that result in best control performance. In the following section, a PBF process control case study is presented, in which such investigation is conducted, in order to gain more intuition on the impact of the radial squashing tuning on RL control performance.

4.5 Simulation results

In this section, a case study of constrained RL for PBF process control is presented. The goal in the specific case study is to adjust the power (action) in order to achieve the desired temperature, while also achieving smooth power changes (action-based constraints). The SAC algorithm is selected for RL control and radial squashing is used for constraint satisfaction, as the final layer in the SAC actor network. Multiple experiments are conducted to investigate the effect of different K values in RL training and control performance, while zero constraint violation is guaranteed by radial squashing definition.

4.5.1 Modelling

The PBF model used in this work comprises a 3D extension [85] of the 2D model found in [18]. More specifically, a back and forth track melting strategy is utilised for building cuboid shapes, and the final part's geometry depends on the determined *number of tracks*, the *length of each track* and the total *number of layers*. The model is designed to demonstrate the resulting heat accumulation (temperature increase) among the layers. The power, P, that is used to melt the powder can be adjusted on a layer-by-layer basis, and the average layer temperature, T_{melt}, is observed as a result of the power used. In this case study, the geometry consists of 5 tracks, 5mm each, and a total of 15 layers. The material used is Ti-6Al-4V powder and the desired temperature, as implemented in [85], is T_{target}=2308K.

In an attempt to investigate the model's behaviour with regards to heat accumulation, multiple layers are built, using the constant power value of P=250W. For the first layer, it is observed that the temperature of 2308K is achieved. However, it is noted that heat accumulates on the next layers, resulting in higher temperatures until saturation, see Figure 4.5. Hence, a control framework is needed in order to manipulate the power on a layer-by-layer basis and achieve the desired temperature of 2308K in every layer.

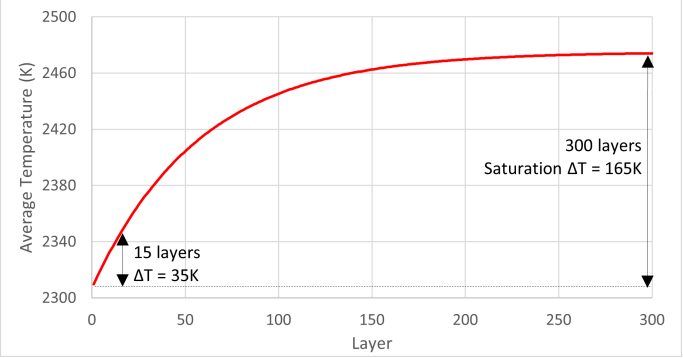

Fig. 4.5 Average layer temperature graph for representation of heat accumulation among the layers. Constant power of P=250W used.

4.5.2 Control formulation

In RL terms, the completion of one part (15 layers) indicates the termination of one episode (15 timesteps). In this case study, the RL algorithm, SAC, is set to be the controller for layer-by-layer PBF process control, initially without constraint handling. Then, SAC is combined with the radial squashing method, as defined in

the methodology section, in order to satisfy the desired boundary constraints. The RL control framework is designed as follows:

- Action space: Continuous space consisting only of the power, 200W$\leq P \leq$300W.
- State space: Continuous space consisting of the observed average layer temperature, T_{melt}, the power, P, that was used to achieve this temperature, and the part's current height (layer).
- Reward function: Based on the work of [14], the reward function r, per layer, is formulated as:

$$r = 1 - \left| \frac{T_{target} - T_{melt}}{100} \right| \tag{4.5}$$

4.5.3 No constraint handling

As a first attempt, the SAC agent is trained without introduction of constraints. 10 separate experiments are run and Figure 4.6 depicts the training process. The mean reward and the standard deviation are shown by the dense line and the error bars respectively. It is observed that the agent achieves high levels of reward, since it reaches a mean maximum of 13.35 out of 15 (maximum theoretically possible), hence 89% training performance.

The resulting policy of the agent and the achieved temperature are shown in Figures 4.7 and 4.8 (average of the aforementioned 10 runs). It is observed that the agent follows a reasonable policy of decreasing power in order to achieve the constant target temperature. Following this policy, the SAC agent achieves a mean temperature of 2308.2K, with a mean absolute error of 0.4K. Hence, the SAC agent successfully addresses the heat accumulation issues.

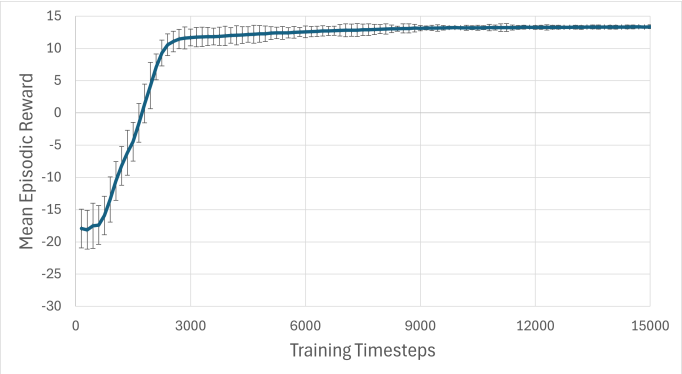

Fig. 4.6 Training curve of the SAC agent. No constraint handling.

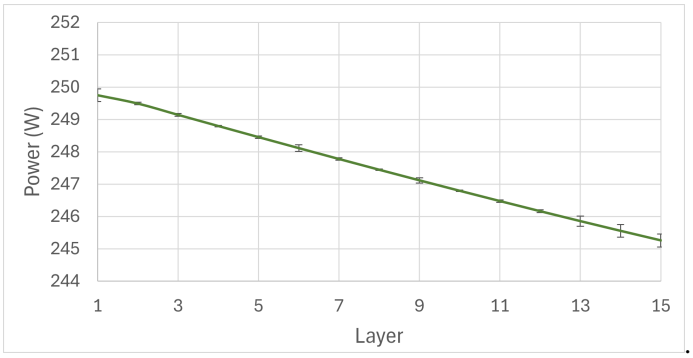

Fig. 4.7 Resulting policy of the SAC agent. No constraint handling.

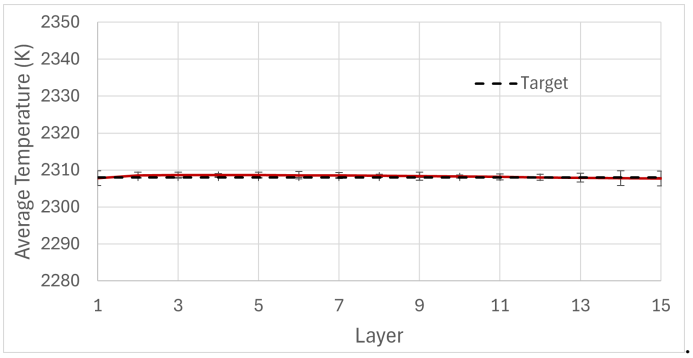

Fig. 4.8 Achieved average temperature of the SAC agent. No constraint handling.

4.5.4 Constraint handling

During the training of the SAC agent in Figure 4.6, there is no constraint introduction, hence the agent can explore freely within the 200W$\leq P\leq$300W limits for a suitable policy. Moreover, as observed in Figure 4.7, the absolute power changes from layer to layer are consistently greater than 0.2W and lower than 1W. Thus, in an attempt to stress-test the controller with regards to constraint handling, boundary constraints of $\pm$0.2W and $\pm$1W are introduced, and the radial squashing method, as discussed in the methodology section, is combined with the SAC agent in order to satisfy these constraints. Finally, in order to test the effect of radial squashing mapping in the SAC training and control performance, the same experiment is repeated for different K values. Following Algorithm 2 (this time, sampling multiple random action trajectories), the effective K region for the $\pm$0.2W constraint is [0.0001, 0.5108], while $K = 0.0023$ is found to be the value that gives *good mapping*. For the $\pm$1W constraint, the effective K region is [0.0004, 2.8381], while $K = 0.0117$ is found to be the value that gives *good mapping*.

Firstly, the SAC agent is trained with the boundary constraint of $\pm$0.2W, and Figure 4.9 depicts the training process for three different K values. It is observed that, in all cases, the agents achieve high levels of reward, since they reach a mean maximum higher than 12.43 out of 15 (maximum theoretically possible), hence higher than 82% training performance. Finally, it is noted that the agent with $K = 0.0001$ is outperformed

by the agents with $K = 0.0023$ and $K = 0.5108$, and this is reflected in the following resulting policies and control results, as well.

The resulting policy of the agents and the achieved temperature for the constraint of ± 0.2W are shown in Figures 4.10 and 4.11. For $K = 0.0001$, which results in a mapping that saturates the attempted actions to the previously taken ones, it is observed that the agent follows a poor policy, keeping the power approximately constant, as expected for such *low* K value. For $K = 0.0023$, which results in a *good mapping* of the attempted actions to the feasible region, it is observed that the agent follows a reasonable policy of decreasing power in order to achieve the constant target temperature. Specifically, it reaches a mean temperature of 2309.5K with a mean absolute error of 6.6K. Finally, for $K = 0.5108$, which results in a mapping that saturates the attempted actions to the boundaries of the feasible region, it is observed that the agent follows a policy of constant +0.2W or -0.2W power changes. By choosing the latter in most timesteps, the agent achieves the best control performance, reaching a mean temperature of 2306.5K with a mean absolute error of 3.7K. Table 4.1 shows the control performance of the SAC agents for different K values, for the ± 0.2W boundary constraint.

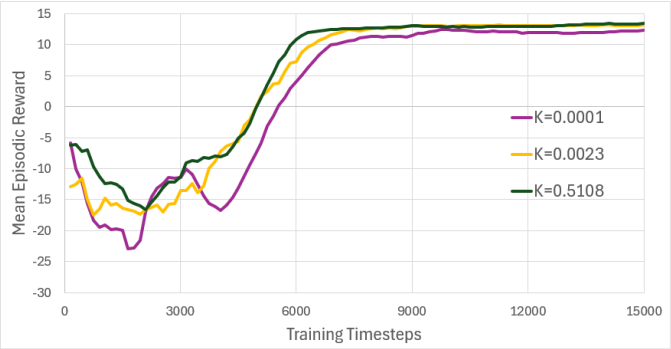

Fig. 4.9 Training curve of the SAC agents. Boundary constraint set at ± 0.2W. Best trained RL agent out of 10 experiments for each K parameter.

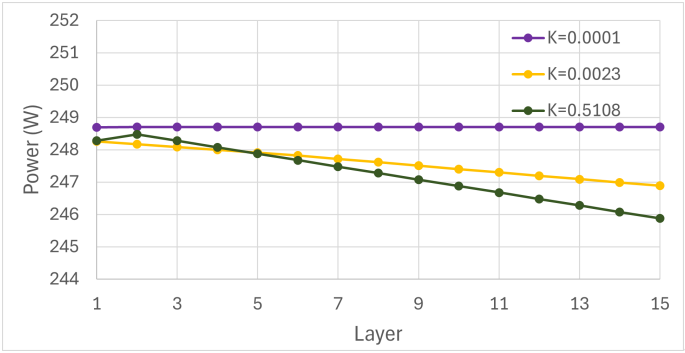

Fig. 4.10 Resulting policy of the SAC agents. Boundary constraint set at ± 0.2W. Corresponding to the best trained RL agent out of 10 experiments for each K parameter.

In order to complete the controller testing with regards to constraint handling, the SAC agent is now trained with the boundary constraint of ± 1W, and Figure 4.12 depicts the training process for three different K values. It is observed that, in all cases, the agents achieve high levels of reward, since they reach a mean

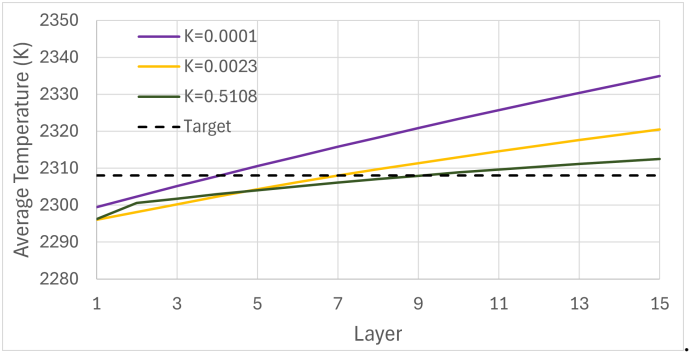

Fig. 4.11 Achieved average temperature of the SAC agents. Boundary constraint set at $\pm$0.2W. Corresponding to the best trained RL agent out of 10 experiments for each K parameter.

Table 4.1 Control performance of the SAC agents for different K values after training completion. Boundary constraint set at $\pm$0.2W

Parameter K	Mean Temperature	Mean Absolute Error
0.0001	2317.9K	12.1K
0.0023	2309.5K	6.6K
0.5108	2306.5K	3.7K

maximum higher than 12.49 out of 15 (maximum theoretically possible), hence higher than 83% training performance. Finally, it is noted that the agents with $K = 0.0004$ and $K = 2.8381$ are outperformed by the agent with $K = 0.0117$, and this is reflected in the following resulting policies and control results, as well.

The resulting policy of the agents and the achieved temperature for the constraint of $\pm$1W are shown in Figures 4.13 and 4.14. For $K = 0.0004$, which results in a mapping that saturates the attempted actions to the previously taken ones, it is observed that the agent follows a poor policy, keeping the power approximately constant, as expected for such *low* K value. The agent, in this case, performs similarly to the previous case with *low* K value (boundary constraint $\pm$0.2W and $K = 0.0001$). Intuitively, it is expected to observe similar policies in both these cases, since the agent's control policy is dominated by the first taken action, regardless of the environment's dynamic behaviour. For $K = 0.0117$, which results in a *good mapping* of the attempted actions to the feasible region, it is observed that the agent follows a reasonable policy of decreasing power in order to achieve the constant target temperature, achieving the best control performance. Specifically, it reaches a mean temperature of 2310.2K with a mean absolute error of 2.2K. Finally, for $K = 2.8381$, which results in a mapping that saturates the attempted actions to the boundaries of the feasible region, it is observed that the agent follows a policy of constant +1W or -1W power changes. Utilising this policy, the agent does not achieve satisfactory performance, reaching a mean temperature of 2303.3K with a mean absolute error of 7.9K. Table 4.2 shows the control performance of the SAC agents for different K values, for the $\pm$1W boundary constraint.

As it is observed in Figure 4.7, under no constraint introduction, the SAC agent's best policy consists of actions with absolute power changes from layer to layer being consistently greater than 0.2W and lower than 1W. Thus, when imposing the $\pm$0.2W boundary constraint, a mapping that leads to action saturation towards

the boundaries ($K = 0.5108$) assists the agent to reach satisfactory control performance, as shown in Table 4.1. However, when imposing the $\pm$1W boundary constraint, a mapping that leads to saturation either towards the center or the boundaries of the feasible region does not assist the agent, since the best power changes do not correspond to the feasible region's center or boundaries. In this case, a K value that results in a *good mapping* ($K = 0.0117$) is preferred, as shown in Table 4.2.

Although the policy in Figure 4.7, under no constraint introduction, is also theoretically achievable in the $\pm$1W boundary constraint case, the SAC agent ($K = 0.0117$) does not manage to learn this policy, and it achieves worse control performance. This phenomenon highlights the known weaknesses of RL control regarding performance guarantees [85], since the achieved policy and control performance highly depend on the RL training, while there are no theoretical performance guarantees.

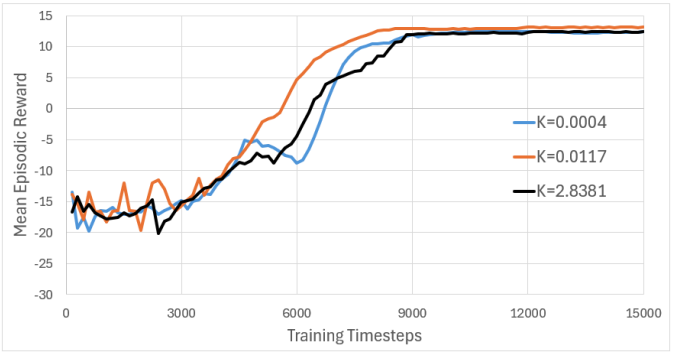

Fig. 4.12 Training curve of the SAC agents. Boundary constraint set at $\pm$1W. Best trained RL agent out of 10 experiments for each K parameter.

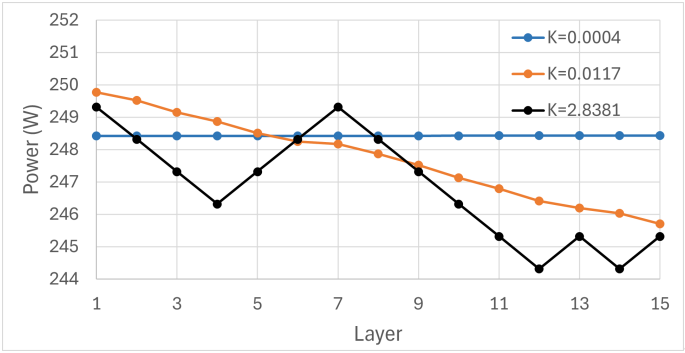

Fig. 4.13 Resulting policy of the SAC agents. Boundary constraint set at $\pm$1W. Corresponding to the best trained RL agent out of 10 experiments for each K parameter.

Moreover, in terms of RL training, when compared with the no constraint case in Figure 4.6, the constraint cases present a deterioration in training performance, especially in the early training stages. This behaviour is expected, since the introduction of constraints can make the portrayed environment more complex for the agent, and potentially further challenge the RL training process. However, it is argued that further investigation on the causes of this deterioration is needed.

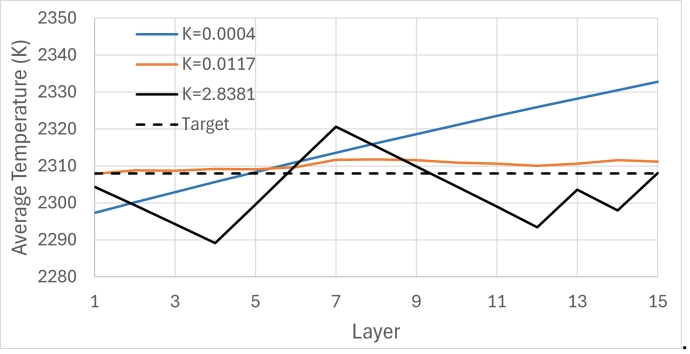

Fig. 4.14 Achieved average temperature of the SAC agents. Boundary constraint set at $\pm$1W. Corresponding to the best trained RL agent out of 10 experiments for each K parameter.

Table 4.2 Control performance of the SAC agents for different K values after training completion. Boundary constraint set at $\pm$1W

Parameter K	Mean Temperature	Mean Absolute Error
0.0004	2315.7K	11.2K
0.0117	2310.2K	2.2K
2.8381	2303.3K	7.9K

4.6 Discussion and future work

In this work, we present a thorough analysis on constrained RL for PBF process control. The radial squashing technique is implemented in the RL framework for action-based constraint satisfaction in PBF, while a hyperparameter is also introduced in order to adjust the intensity of the squashing (tuning). Multiple experiments are conducted on a PBF simulation platform, under different constraints, assessing the impact of the radial squashing tuning on control performance. The results show that adjusting the intensity of the squashing is necessary for maintaining satisfactory control performance under constraints.

Although the heuristic experimentation with the hyperparameter, K, can result in satisfactory RL control performance, it is arguably suboptimal, since it depends on which and how many different K values are tested. Moreover, this heuristic experimentation refers only to implementations in which the K value remains constant during each training process and implemented control policy. However, changes of the K value from episode to episode, or from timestep to timestep, should also be investigated, since, for certain PBF case studies, the key to best control can be the in-process relaxation/intensification of the imposed constraint. Literature evidence, such as [81] and [85], supports the above arguments, since it presents attempts to tune (or auto-tune), in-process, such hyperparameters that affect the RL training and control performance.

As a result, we encourage further research with focus on the tradeoff between feedback control effectiveness and constraint handling via the introduced hyperparameter, K, perhaps as an optimisation problem.

4.7 Funding

This study was funded by the UK Engineering and Physical Sciences Research Council (EP/P006566/1) and iCASE (EP/T517835/1), with contributions from Wayland Additive Ltd.

Chapter 5

Bridging simulation and practice: reinforcement learning for electron beam melting control

The main challenge of the constrained RL method described in the previous chapter is the reliance on the manually tuned intensity hyperparameter K. While experimental results demonstrate that careful manual tuning of K can lead to satisfactory control performance, the process is inherently suboptimal, and sensitive to the specific choice of the K value throughout RL training and control. This approach lacks the adaptability needed for real-world PBF manufacturing environments, in which process conditions can evolve rapidly and demand dynamic constraint management.

To overcome this limitation, this chapter introduces an auto-tuned, constraint-aware RL framework. Instead of relying on fixed K values, the proposed method enables in-process modulation of constraint intensity. This shift from manual to automated constraint adjustment marks a critical development in safe RL for manufacturing, supporting more flexible and responsive control strategies suitable for complex, dynamic scenarios such as PBF applicatons, as shown in this chapter, section 5.5.

In addition to this key novelty point, this chapter presents a substantial step forward in bridging theory and practice. A computationally efficient, real-world based PBF simulation framework is introduced in this chapter, section 5.3.2, representing the class of EBM manufacturing. The developed simulation platform can capture meltpool behaviour on both simple and overhang geometries.

Using this platform, RL control strategies are tested across multiple layers and geometrical scenarios. Particular emphasis is placed on constraint satisfaction and noise robustness, critical qualities for any RL system to be deployed on real-world manufacturing equipment. This chapter concludes with a comprehensive evaluation of RL performance under these realistic conditions, highlighting both the strengths and current limitations of the tested approaches.

Note: the presented paper is slightly amended for the purposes of this thesis. The original paper has been peer reviewed and accepted in the Journal of Manufacturing Processes, 2025: Stylianos Vagenas, Nicholas Boone, and George Panoutsos. Bridging simulation and practice in additive manufacturing: Reinforcement learning for electron beam melting control. Manufacturing Processes, 2025. Elsevier.

https://doi.org/10.1016/j.jmapro.2025.11.026

5.1 Abstract

Reinforcement Learning (RL) continues to show great promise for intricate process control applications, particularly in areas in which the application model is too complex to design and conventional control methods may fall short. Powder Bed Fusion (PBF) processes, such as Electron Beam Melting (EBM), are prime examples. These processes involve highly dynamic and nonlinear physical phenomena that are challenging to capture in control-oriented models. RL offers a data-driven, flexible alternative. However, its practical implementation remains limited by a lack of real-world based testing environments. To bridge this gap, our work includes the development of computationally efficient, real-world based simulation models of the EBM process, which serve as a critical platform for testing RL control strategies under realistic conditions. The real-world based models capture the essential thermal process dynamics, both for simple cuboids, as well as for overhang structures, while maintaining low computational cost, hence being suitable for iterative training and evaluation of RL algorithms. We use these models to implement and assess RL control across multiple layers, demonstrating RL's ability to manage key challenges such as meltpool control. Importantly, this work provides the final experimental step before RL deployment on a physical machine, ensuring that the RL strategies tested are both effective and safe under conditions that closely mimic the real operational environment. By incorporating considerations of satisfactory performance and RL constraint satisfaction, we demonstrate that RL can achieve reliable performance even in safety-critical EBM settings. Our results underscore the importance of realistic simulation platforms for advancing RL in industrial control, and open a direct pathway towards its adoption in real-world manufacturing systems.

5.2 Introduction

Powder Bed Fusion (PBF) is a prominent manufacturing technique that enables the fabrication of intricate, high-performance parts directly from digital designs. Leveraging layer-by-layer deposition and selective melting, PBF offers advantages such as geometric freedom, reduced tooling, and the ability to engineer tailored microstructures, see [56] and [71]. However, the complex, multi-scale physical phenomena involved in the PBF process, and particularly in Electron Beam Melting (EBM), make it difficult to model and control effectively, especially in the presence of geometrical features such as overhangs, which significantly affect heat distribution, meltpool behaviour, and solidification dynamics, see [12], [90] and [91].

Despite progress in PBF simulation modelling and control, see [18], [14] and [85], most of the existing models in the literature are either unrealistic versions of the actual PBF process, or they are not formulated to support control development. This leads to a critical gap: the absence of control-oriented, real-world based models that can accurately represent key process dynamics, such as meltpool shape/area evolution, while being computationally efficient enough to support learning and feedback control. Importantly, such models must also be able to simulate varying geometrical complexities, including simple and overhang structures, which are commonly encountered in real-world parts and present unique control challenges, see [12] and [92].

To address this gap, this work contributes computationally fast, control-oriented simulation models of the EBM process that are designed for direct use in the development and evaluation of advanced control strategies. The models capture the essential dependencies relevant to meltpool morphology, while remaining fast enough for iterative optimisation. Moreover, they also support simulation of varying part geometries, including

overhangs, enabling robust testing of control frameworks under realistic manufacturing scenarios. Building upon these models, Reinforcement Learning (RL) control frameworks are applied and evaluated, which are capable of adapting to the process dynamics over multiple layers. RL offers a promising solution for EBM control, given its model-independent nature and ability to learn from interaction rather than requiring explicit physical formulations or strict signal assumptions. This is particularly beneficial for parts with complex geometries or dynamically evolving meltpool profiles, in which traditional feedforward or PID controllers may be insufficient, see [85].

Crucially, an additional focus of this work is on constraint-aware RL control. In the EBM process, violating process constraints, such as abrupt dwell time changes or exceeding energy thresholds, can degrade part quality or damage equipment. Hence, it is essential to maintain a zero constraint violation framework throughout the manufacturing process. In order to provide that framework, safe RL control agents need to be developed, capable of real-world machine deployment. The work of [93] provides such constrained RL frameworks, in which the intensity of the imposed constrained is manually tuned. In this work, this tuning is automated, hence an auto-tuned, constraint-aware RL control framework is implemented.

In conclusion, the purpose of this work is to develop and evaluate RL frameworks based on real-world based, control-oriented models of the EBM process. The models represent the multi-layer EBM process, both for simple cuboid, as well as for overhang structures. The openloop results show that the absence of a controller leads to meltpool area issues, hence process control is considered. RL is proposed as a control method to adjust the dwell time of the heat source based on process monitoring and feedback, while also taking dwell time constraints into account to guarantee safe deployment. The main contributions of this work are the following:

- Development of a real-world based EBM simulation platform, and implementation of RL process control.
- Implementation of RL process control for simple and complex control objectives (varying target), including noisy signals.
- Demonstration of the benefits and the limitations of RL process control on simple cuboid geometries, as well as on more complex, overhang structures.
- Assessment of auto-tuned, constraint-aware, RL control strategies, resulting to agents which are safe for real-world deployment.

5.3 EBM modelling and control

5.3.1 The EBM process

EBM manufacturing machines, part of the electron beam PBF family, use metallic powders such as titanium alloys as feedstock. In a typical EBM setup, the metalic powder is stored in a platform and it is spread across the build plate using a recoating mechanism. An electron beam, operating under vacuum conditions, then selectively scans and melts regions of the powder layer according to a predetermined pattern. After each layer is processed, the build plate is lowered, and a new layer of powder is spread. This cycle is repeated iteratively

until the final component is completed, while the whole manufacturing process is usually monitored by an IR camera. Figure 5.1 shows a schematic representation of the machine layout, including the aforementioned features and the supports, which are often used in order to assist the manufacturing of overhang structures.

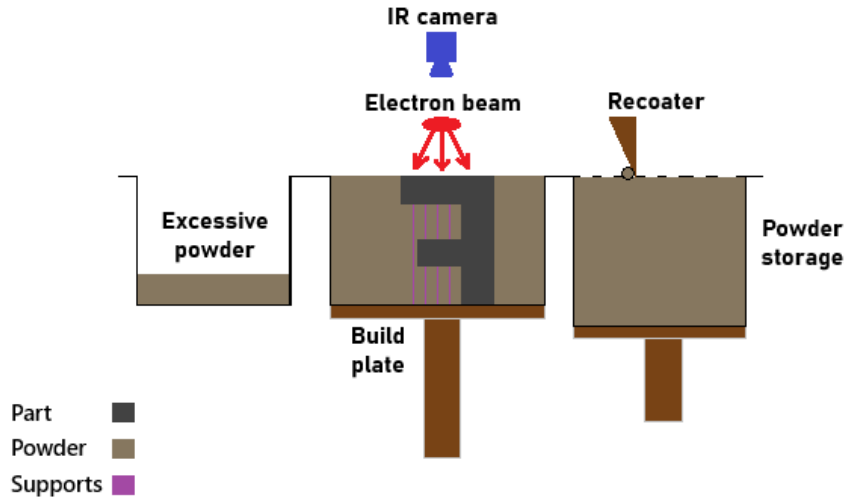

Fig. 5.1 Schematic of an EBM machine set up.

While extensive studies have explored the fundamental physics of EBM, much of the work has focused on understanding and controlling meltpool behaviour, as it directly influences the resulting microstructure and mechanical properties of the manufactured part. The meltpool, formed where the electron beam interacts with the powder, plays a critical role in determining grain morphology, porosity levels, and final part density. EBM, operating in high preheating temperatures and under vacuum, shows significantly different thermal gradients and solidification behaviour compared to other PBF processes.

In EBM, maintaining a consistent and controlled meltpool area, is key to achieving desirable and repeatable microstructural features. However, this becomes particularly challenging when manufacturing parts with complex geometries, such as overhangs, as opposed to simpler structures like cuboids. Overhangs often experience uneven heat dissipation and altered beam-powder interactions, which can lead to irregular meltpool shapes/areas, and increased defect likelihood due to inadequate support and dissipation beneath the structure. In this work, both cuboids and overhang structures without supports are investigated, and the meltpool area is chosen as the primary objective.

5.3.2 Developing a real-world based EBM model

As previously mentioned, an important gap in the literature is regarding the absence of control-oriented, real-world based models. In this work, in order to develop a real-world based EBM model, real-world builds are implemented with the Calibur3 machine (Wayland Additive Ltd.), which is shown in Figures 5.2 and 5.3, both for cuboids and for overhang structures. More specifically, 6 cuboids with 10 mm x 10 mm base and 856

layers tall are built, and 6 overhang structures are built with the same dimensions, including 4 balcony regions without supports, as in Figure 5.4. All 6 cuboids are built on the same thermal conditions, with the dwell time varying randomly within 160μs and 200μs, which is the operating window for the EBM machine. All 6 overhangs are built on the same thermal conditions, with the same dwell time variation as the cuboids. The material used is Ti-6Al-4V powder. During the manufacturing of all the parts, the parts' thermal behaviour is monitored and the meltpool area is captured via an IR camera.

The image processing stages are given in Figures 5.5 and 5.6 as the top view of the build platform. For these example frames, as well as for every frame during the manufacturing process, the regions of interest are defined. Then, the hot spots are identified, and the hot spot that comprises the actual meltpool area is stored as a result for this frame. The rest of the appeared hot spots are in fact meltpools from previous frames that are cooling down. As a result, for every dwell time used on every layer on each of the 12 parts, the resulting average meltpool area is stored per layer per part.

Fig. 5.2 The Calibur3 manufacturing machine, Wayland Additive Ltd.

After the IR image processing and the storing of the data, two Neural Networks (NNs) are trained to be the EBM models. Due to the nature of the EBM process, the effect of the thermal history (thermal behaviour on a single layer resulting also from multiple layers underneath) and complex time dependencies, the Long Short-Term Memory NNs are trained, called LSTMs. The LSTMs are well-established NNs and they are able to identify and capture such dependencies, see [94]. The LSTM models' inputs are the current number of layer and the dwell time used. The models' output is the resulting average meltpool area on this layer. The training and validation graphs for the LSTM model for the cuboid and the LSTM model for the overhang are given in Figure 5.7.

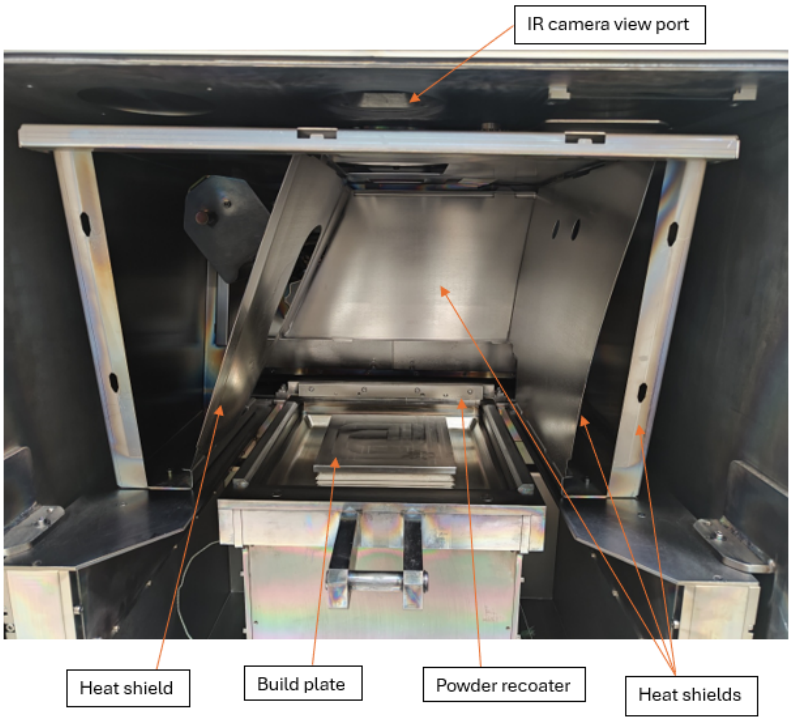

Fig. 5.3 The Calibur3 build chamber, Wayland Additive Ltd.

It is observed that the training for both models is successful, since the training loss and validation loss have converged to a minimum. The validation loss is slightly higher than the training loss, as expected, since it refers to the models being validated/tested in not-before-seen data.

The training and the validation of the LSTMs result to two real-world based EBM models. Each time the models are called, they expect the number of layer and the dwell time as inputs, and as a result they give the average meltpool area on this layer. Hence, from the model's perspective, each call comprises a timestep, and it corresponds to the build of a whole layer. Importantly, LSTM models can suffer from low accuracy during the first timesteps, either due to lack of context or due to cold start effects, as discussed in [94]. Hence, the models' predictions are considered to be accurate after some layers have already been built.

In this work, in order to save computational time, while still maintaining the generality of the resulting conclusions, the models are used to build two specific geometries: a simple cuboid, 200 layers tall, and an overhang structure with the same height, including two overhang balcony regions of 25 layers each, as in Figure 5.8. In Figure 5.9 the openloop behaviour of these two geometries is shown, built with the nominal dwell time value of 180μs. The models' predictions are considered to be accurate after the 50th layer.

It is observed that both models demonstrate the expected behaviour in accordance to the geometry built. The cuboid shows a gradually increasing meltpool area under constant dwell time, as heat accumulates and results in larger meltpool regions, see relevant work of [85]. The overhang structure shows poor heat dissipation in the overhang balcony regions, see relevant work of [12] and [95], resulting to large meltpool peaks in temperature and area metrics. Hence, in both geometries, effective control methods need to be developed in order to address the meltpool area issues and achieve the desired meltpool area targets.

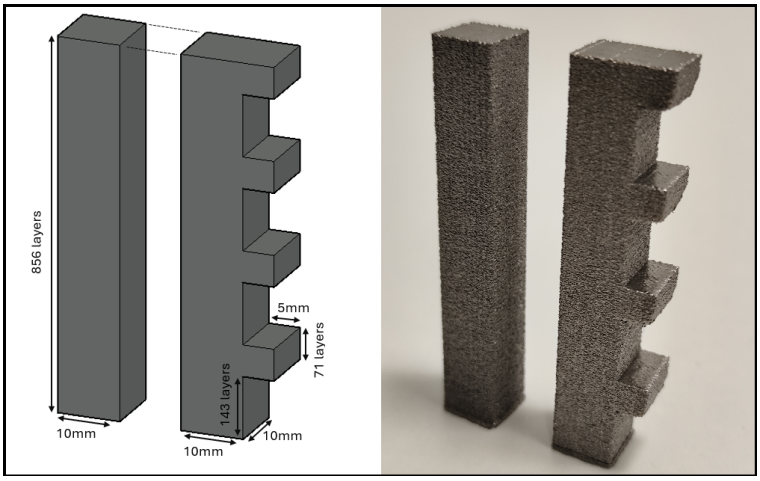

Fig. 5.4 3D illustration of the two geometries built in Calibur3 machine. 3D design on the left, and the actual part built on the right.

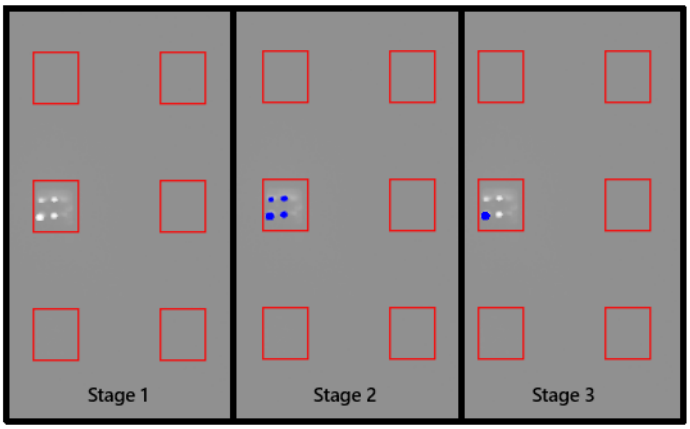

Fig. 5.5 Image processing for the cuboid geometry. Regions of interest surrounding the cuboid build positions. Example frame top view.

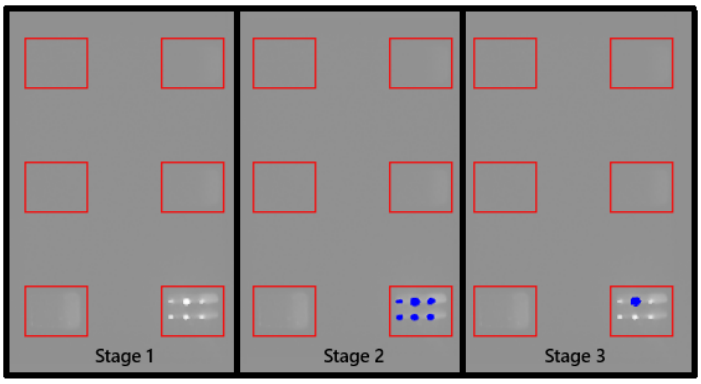

Fig. 5.6 Image processing for the overhang geometry. Regions of interest extended in order to capture the overhang balconies. Example of an overhang balcony frame top view.

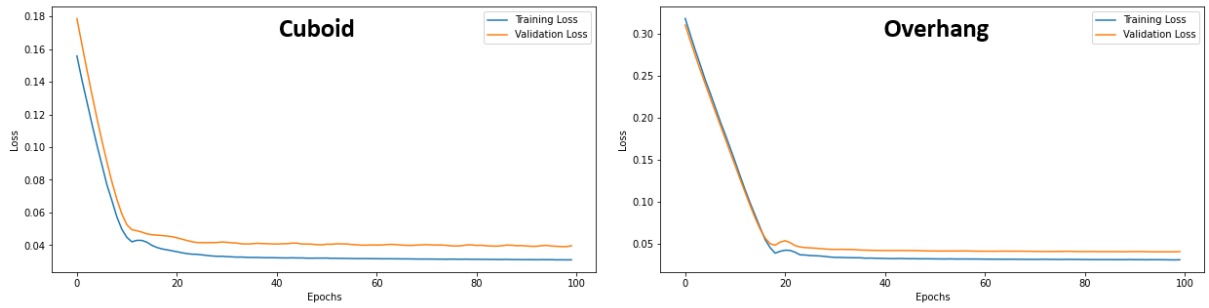

Fig. 5.7 Training and validation graph of the LSTM models. 60 nodes used for the cuboid LSTM and 100 nodes for the overhang LSTM, ReLU activation function in both cases, 20% validation split.

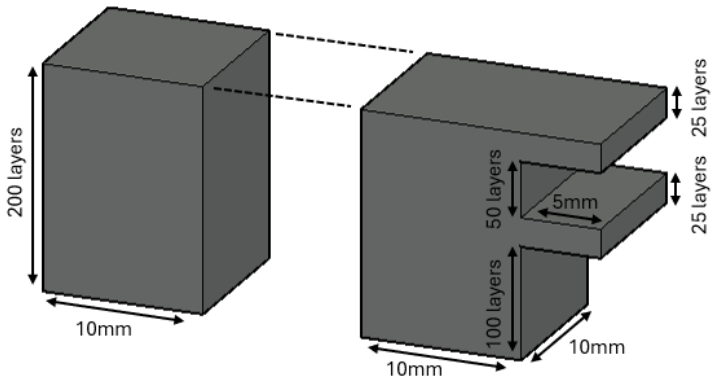

Fig. 5.8 3D representation of the two geometries built using the resulting LSTM models.

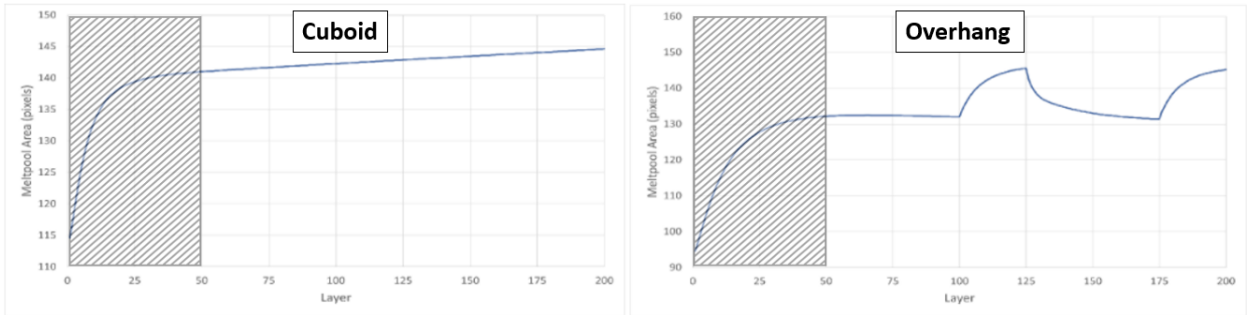

Fig. 5.9 Openloop behaviour of the two geometries in Figure 5.8, resulting from the two LSTM models. Both parts built with constant dwell time value of 180μs. Models' predictions are considered to be accurate after the 50th layer.

5.3.3 Process control in EBM

Process control in EBM is a crucial requirement for achieving consistent quality, especially as part geometry complexity increases and the desired microstructure becomes more intricate. Unlike simple geometries, such as solid cuboids, within which thermal behaviour and meltpool dynamics are relatively stable and easier to manage, overhang structures and complex geometries can introduce thermal instabilities and unpredictable variations in meltpool areas, see [12]. These challenges make openloop strategies insufficient and underscore the need for robust process control systems to ensure the desired microstructure, and avoid process defects. Process control becomes even more critical when the control target varies among layers, see [85], or when noise/disturbance is introduced into the process.

Traditionally, control theory based methods, such as PID or feedforward control, have been applied in PBF due to their simplicity and industrial familiarity, see [12], [28], [30] and [76]. However, their effectiveness can prove to be limited in high-complexity PBF tasks, as shown in [85], due to action delays and offsets from the control target, see [75]. More recently, data-driven methods, including RL, have emerged as a promising alternative. RL can learn control policies from data, potentially adapting to system variability and unmodelled dynamics. For this reason, this work focuses on the implementation and assessment of RL control frameworks. The control formulation, along with the control targets and the design of the RL agents are discussed in the following sections.

5.4 Reinforcement learning for process control

5.4.1 Reinforcement learning overview

RL is an iterative, data-driven method that enables an agent (controller) to learn optimal actions through interaction with the environment. Rather than being explicitly instructed with specific rules, the agent uses trial-and-error to discover strategies that maximise a defined reward signal, see [40]. A distinctive feature of RL is that it does not only focus on maximising the reward in an immediate sense, but on achieving high rewards cumulatively, in the long run.

The RL process begins on timestep, t, with the agent in a state, s_t, from which it selects an action a_t. This action affects the environment, which responds by providing a new state, s_{t+1}, and a corresponding reward, r_{t+1}. For instance, within the PBF domain, the action might be the dwell time of an EBM heat source, while the reward might relate to maintaining a consistent meltpool area.

A widely used and effective RL approach for process control is the actor critic method, see [41], [42], [43]. In Figure 5.10, a generic structure is shown for the actor critic class of RL control. The agent is formulated as a pair of neural networks, which play the role of function approximators. The policy function approximator is denoted with $\pi_\theta(s,a)$ (actor network), parameterised by θ, and the value function approximator is denoted with $v_w(s)$ (critic network), parameterised by w.

For this work, the Soft Actor Critic (SAC) algorithm is used as proposed in [44] and [81]. The SAC algorithm is a model-free actor critic method within the RL domain. It aims not only to maximise expected rewards but also to maximise entropy (via entropy regularisation), which enhances the exploration of control actions. As proposed in [81], the entropy regularisation parameter is auto-tuned, initiating in explorative behaviour (chance of low rewards in the early steps of training), and progressively reaching more exploiting

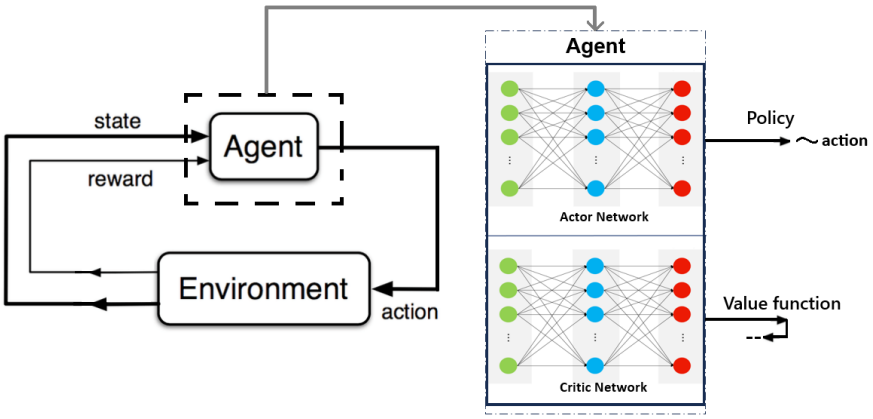

Fig. 5.10 The RL paradigm in the actor critic class of methods.

behaviour (prioritising high rewards). Regarding the agent's structure, it consists of an actor neural network, $\pi_\theta(s,a)$, responsible for policy updates, determining action selection, and two critic neural networks, $v_1 w(s)$ and $v_2 w(s)$, responsible for evaluating the quality of the taken action. The reason behind the two critic neural networks is to address potential overestimating issues of the value function, as the minimum estimated value function between the two networks, $min(v_1 w(s), v_2 w(s))$, is considered to be the valid one. Considered as a state of the art RL algorithm, SAC has demonstrated exceptional performance in continuous control benchmarks and more complex tasks, see [43] and [85].

5.4.2 Constrained reinforcement learning

In SAC, it is standard practice to define overall action limits (such as dwell time) in continuous control settings, as demonstrated in all benchmark control tasks, see [86]. Most SAC implementations enforce these limits by applying a hyperbolic tangent function as the final activation layer in the actor network, an approach known as squashing, see [53]. However, this simple squashing method does not accommodate dynamic, in-process constraints. Such constraints are especially important in critical applications, such as EBM, for maintaining part integrity and adhering to machine limitations.

The work of [93] provides a method to apply action-based constraints (action constraints based on previous actions taken) in SAC process control, by adopting a radial squashing variant. More specifically, this radial squashing variant is described as follows. Let a_t be the taken action on timestep t. Based on the taken action a_t, the boundaries b_1 and b_2 are defined as constraint for the next action, with $b_1 < a_{t+1} < b_2$. If the next attempted action by the agent is $a'_{t+1} < a_t$, then the update rule in (5.1) is applied to satisfy the constraint, whereas if $a'_{t+1} > a_t$, then the update rule in (5.2) is applied. As a result, this attempted action is squashed into the feasible region, $[b_1, b_2]$, and the actual taken action, a_{t+1}, satisfies the constraint. Following this reasoning, the action a_{t+1} determines the new boundaries for the next action, a_{t+2}. Hence, every attempted action is mapped into its feasible region, assuring the desired smoothness of transitions from action to action and guaranteeing zero constraint violation on every timestep of the action trajectory.

The exact value of the next taken action depends on the value of the introduced K parameter. Currently, one has to manually experiment with the K parameter in order to find the optimal K value for a specific case

study, as one cannot know beforehand which specific K value (or which K value profile) works best in the interest of the SAC agent, see [93]. In this work, this process is automated by including the K parameter in the action space of the SAC agent as an additional action.

$$a_{t+1} = a_t + tanh\left(K\frac{|a'_{t+1} - a_t|}{|b_1 - a_t|}\right)(b_1 - a_t) \tag{5.1}$$

$$a_{t+1} = a_t + tanh\left(K\frac{|a'_{t+1} - a_t|}{|b_2 - a_t|}\right)(b_2 - a_t) \tag{5.2}$$

5.4.3 Control formulation

The EBM models under investigation are the two LSTM models described in the previous sections, representing a simple cuboid and an overhang structure. For both geometries, the base is a 10 mm x 10 mm square, and the total number of layers built is 200. As previously mentioned, the LSTM method is helpful with regards to capturing timestep dependencies, since this is particularly needed in the EBM domain. However, it suffers from the drawback of low accuracy during the first timesteps (layers). This is taken into account in the upcoming process control attempts, by assuming a window of 50 layers built before activating the controller. More specifically, in each control case of the following section, the first 50 layers of each part are built with the nominal dwell time value of 180μs, and from layer 51 onwards the designed controller is activated and the EBM behaviour and the relevant metrics are captured. The two geometries can be observed in Figure 5.8 and the openloop behaviour under constant dwell time can be observed in Figure 5.9.

From the RL agent's perspective, each layer build is considered to be a process timestep, and the completion of the whole geometry is considered to be an episode. The controller's action per timestep is the dwell time used for building a specific layer, limited between 160μs and 200μs. The state includes the average meltpool area achieved on this specific layer (timestep). The reward signal, hence, the control target per timestep, is designed with reference to the meltpool area achieved. A_{melt} is defined as the average layer area observed and A_{target}, as the desired layer area. Hence, the reward per timestep is defined as:

$$r = 1 - \left|\frac{A_{target} - A_{melt}}{10}\right| \tag{5.3}$$

More specifically, the SAC controller (SAC) is trained and tested in both the cuboid and the overhang structure for a fixed control target and for a varying control target scenario. Moreover, a noise signal is introduced to the meltpool area observed in order to challenge the SAC agent (SAC-noisy) and better mimic realistic process control conditions. Finally, the SAC-noisy agent is combined with the radial squashing constraint method (CSAC-noisy), see [93], in order to mimic potential machine/material limitations and establish a zero constraint violation framework. A constraint of $\pm 2\mu$s of the previous action is applied on the action to be taken, and the effective K region is calculated to be (0, 5), using the algorithm in [93]. The K values close to 0 saturate the attempted action to the value of the previous action taken, while the K values close to 5 saturate the attempted action to its feasible constraint boundaries. The rest of the K values within the effective K region can provide a more flexible mapping of the attempted action into the feasible constrained action region. Unlike [93], in this work, the tuning K parameter is included in the action space, hence the

original actor network now gives two actions; the attempted dwell time, and the tuning K parameter, which is used to squash the attempted dwell time into the feasible constrained dwell time region, see (5.1) and (5.2), resulting to the final dwell time value applied on the specific layer.

In all case studies, the RL agent is trained for 1800 episodes, each consisting of 150 timesteps, i.e. 270000 training timesteps. 10 separate experiments are run in each case study, for which the results that correspond to the best training process are shown. The RL implementation details for these case studies are presented in Tables 5.1 and 5.2.

Table 5.1 SAC implementation summary

Action space Box(1,)	State space Box(3,)	Episode duration
Dwell time (160, 200)	Meltpool area, Dwell time, Layer	150 timesteps (layers)

Table 5.2 Constrained SAC implementation summary

Action space Box(2,)	State space Box(3,)	Episode duration
Dwell time (160, 200); K (0, 5)	Meltpool area, Dwell time, Layer	150 timesteps (layers)

5.5 EBM process control results

5.5.1 SAC control for cuboid geometry

The first case study is RL process control for the cuboid geometry, with the aim of maintaining the average layer meltpool area constant throughout the build. The desired layer area is set to be A_{target}= 142 pixels, the value equal to the openloop area value observed averaging from layer 51 until layer 200. Moreover, in order to closely mimic the real process, a noise signal of 1% is also introduced in the observed area signal. The RL controller framework is designed as shown in Table 5.1.

Figure 5.11 depicts the best training process of the SAC agent without noise in the area signal, and with 1% noise in the area signal, after 10 training runs for each case. It is observed that the SAC agent outperforms the SAC-noisy agent, as expected. It achieves a maximum reward higher than 147 out of 150 (maximum theoretically possible), hence higher than 98% training performance. On the other hand, the SAC-noisy agent achieves a lower maximum reward of 141.91, however, this is still considered satisfactory. The training metrics are shown in Table 5.3.

Fig. 5.11 Best training curves of the SAC and the SAC-noisy agents. Fixed target for the average layer area, cuboid geometry.

Table 5.3 Fixed target training comparison between SAC and SAC-noisy, cuboid geometry

RL agents	Mean Reward	Maximum Reward
SAC	142.34	147.86
SAC-noisy	139.35	141.91

The resulting policy and the achieved average meltpool area in each layer are shown in Figures 5.12 and 5.13. It is observed that the SAC agent demonstrates exceptional performance, managing to maintain the average layer area in the desired value with a Mean Absolute Error (MAE) of 0.05 pixels. Moreover, despite the noise, the SAC-noisy agent also manages to pick up a suitable policy, with a similar trend, and it manages to maintain the average layer area in a satisfactory manner throughout the build. Overall, it is argued that in both cases, the heat accumulation issues observed in the corresponding openloop results of Figure 5.9 are sufficiently addressed. The result metrics for the controllers are shown in Table 5.4.

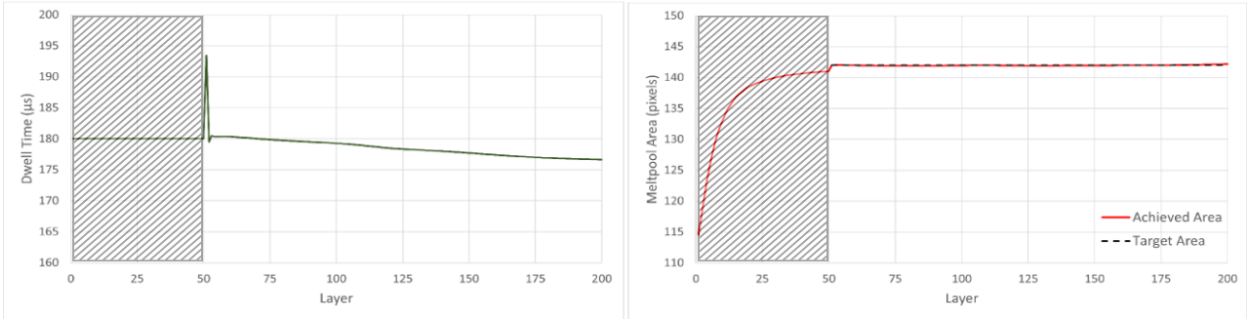

Fig. 5.12 Resulting policy of the SAC agent with fixed target, cuboid geometry.

In order to test the agents, the control problem is now altered by introducing a varying A_{target}. The research interest in varying target applications is not only from a control perspective, but also from an EBM perspective, since varying meltpool areas throughout the build can be an interesting EBM objective towards achieving

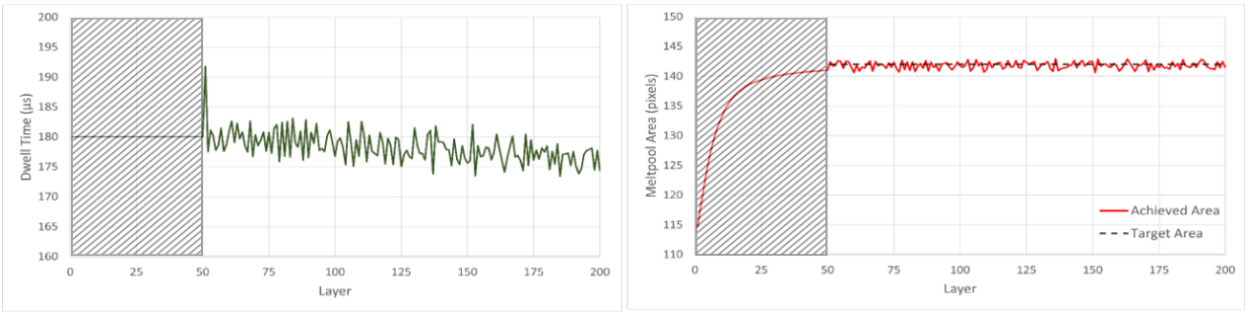

Fig. 5.13 Resulting policy of the SAC-noisy agent with fixed target, cuboid geometry.

Table 5.4 Fixed target results comparison of SAC and SAC-noisy, cuboid geometry

Controller	Mean Area (pixels)	MAE (pixels)
Openloop	142	1.11
Openloop-noisy	142	1.29
SAC	142.15	0.05
SAC-noisy	141.86	0.52

bespoke microstructures. Unlike the work of [85] with the multi-step approach, the varying control target in this work consists of a combination of sine functions, resulting to smooth target changes. Once again, in order to closely mimic the real process, a noise signal of 1% is also introduced in the observed area signal. The RL controller framework is designed as shown in Table 5.1.

Figure 5.14 depicts the best training process of the SAC agent without noise in the area signal, and with 1% noise in the area signal, after 10 training runs for each case. It is observed that the SAC agent outperforms the SAC-noisy agent, as expected. It achieves a maximum reward higher than 143 out of 150 (maximum theoretically possible), hence higher than 95% training performance. On the other hand, the SAC-noisy agent achieves a lower maximum reward of 139.36, however, this is still considered satisfactory. The training metrics are shown in Table 5.5.

Fig. 5.14 Best training curves of the SAC and the SAC-noisy agents. Varying target for the average layer area, cuboid geometry.

Table 5.5 Varying target training comparison between SAC and SAC-noisy, cuboid geometry

RL agents	Mean Reward	Maximum Reward
SAC	134.73	143.37
SAC-noisy	132.34	139.36

The resulting policy and the achieved average meltpool area in each layer are shown in Figures 5.15 and 5.16. It is observed that the SAC agent demonstrates very good performance, managing to track the average layer area in the desired value. Moreover, despite the noise, the SAC-noisy agent also manages to pick up a suitable policy, with a similar trend, and manages to track the average layer area in a satisfactory manner throughout the build. In all, the SAC controller achieves a MAE of 0.34 pixels and the SAC-noisy controller achieves a MAE of 0.65 pixels.

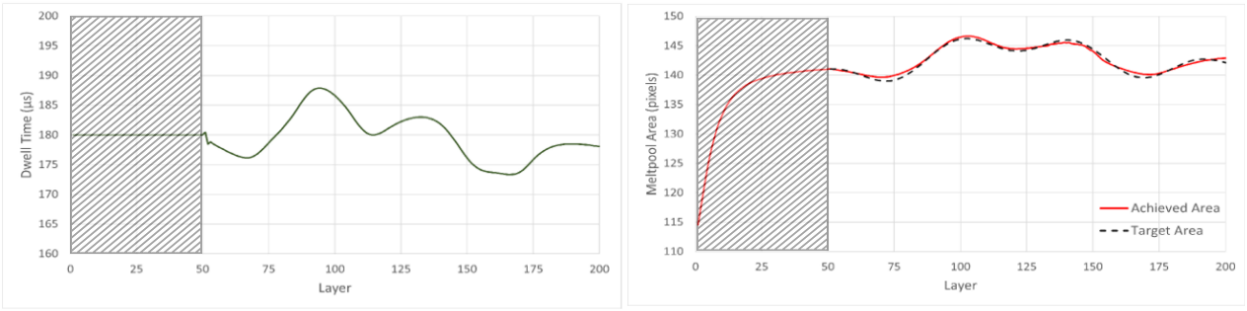

Fig. 5.15 Resulting policy of the SAC agent with varying target, cuboid geometry.

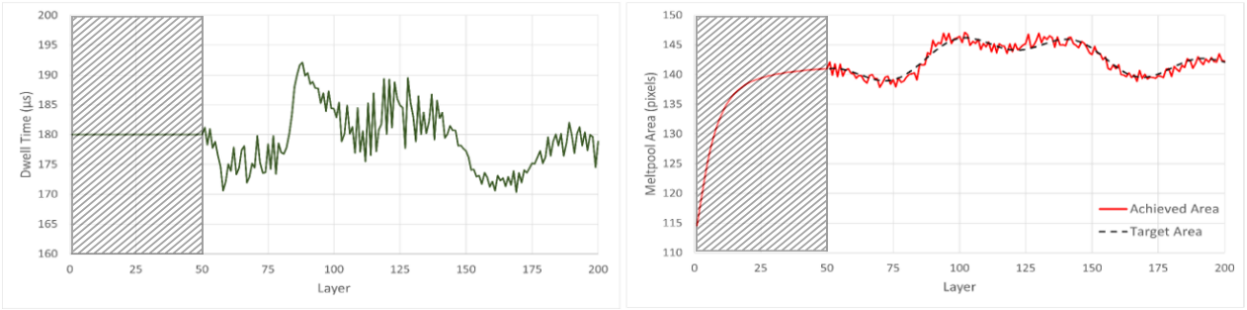

Fig. 5.16 Resulting policy of the SAC-noisy agent with varying target, cuboid geometry.

5.5.2 SAC control for overhang geometry

In the cuboid geometry case, the agent demonstrates satisfactory performance both in a fixed target and in a varying target setting, with and without noise. However, for the more complex, overhang geometry, it is plausible for the agent to be significantly challenged. In order to investigate this argument, RL process control is implemented for the overhang geometry, with the aim of maintaining the average layer meltpool area constant throughout the build. The desired layer area, set to be A_{target}= 137 pixels, the value equal to the openloop area value observed averaging from layer 51 until layer 200. Moreover, in order to closely mimic the

real process, a noise signal of 1% is also introduced in the observed area signal. The RL controller framework is designed as shown in Table 5.1.

Figure 5.17 depicts the best training process of the SAC agent without noise in the area signal, and with 1% noise in the area signal, after 10 training runs for each case. It is observed that the SAC agent outperforms the SAC-noisy agent, as expected. It achieves a maximum reward of 117.20 out of 150 (maximum theoretically possible), hence approximately 78% training performance. The training metrics are shown in Table 5.6.

It is argued that the agents are significantly challenged in the overhang case study with 78% training performance, compared to the correspondent cuboid case study with training performance higher than 98%. For more clarity, the focus of the analysis turns to the resulting policy in order to understand better why the agents are challenged.

Fig. 5.17 Best training curves of the SAC and the SAC-noisy agents. Fixed target for the average layer area, overhang geometry.

Table 5.6 Fixed target training comparison between SAC and SAC-noisy, overhang geometry

RL agents	Mean Reward	Maximum Reward
SAC	103.83	117.20
SAC-noisy	98.44	110.23

The resulting policy and the achieved average meltpool area in each layer are shown in Figures 5.18 and 5.19. In the bulk regions of the part, it is observed that both agents demonstrate very good performance, managing to maintain the average layer area in the desired value. In the overhang balcony regions, the agents attempt to minimise the dwell time value to 160μs in order to compensate for the overhang overheating phenomenon. This technique alone seems insufficient to address the overhang overheating phenomenon (potential need for lower than 160μs dwell time values). Hence, the RL agents, having realised the different dynamic behaviour in the overhang balcony areas, choose to drop the dwell time value on the layer before (layer 100 and layer 175). This results in deviating from the control target for these two layers, but achieving meltpool areas which are overall closer to the target within the overhang balcony region. This behaviour of the RL controller aligns with the RL features described in the previous sections, confirming that RL

focuses on maximising rewards in a cumulative fashion and not just on an immediate sense. Overall, these policies in Figures 5.18 and 5.19 provide a small, but significant improvement in performance compared to the corresponding openloop results in Figure 5.9. However, it is argued that the overhang overheating phenomenon is not sufficiently addressed. The result metrics for the whole build, along with specific results for the overhang balcony regions (Ov.) for the controllers are shown in Table 5.7.

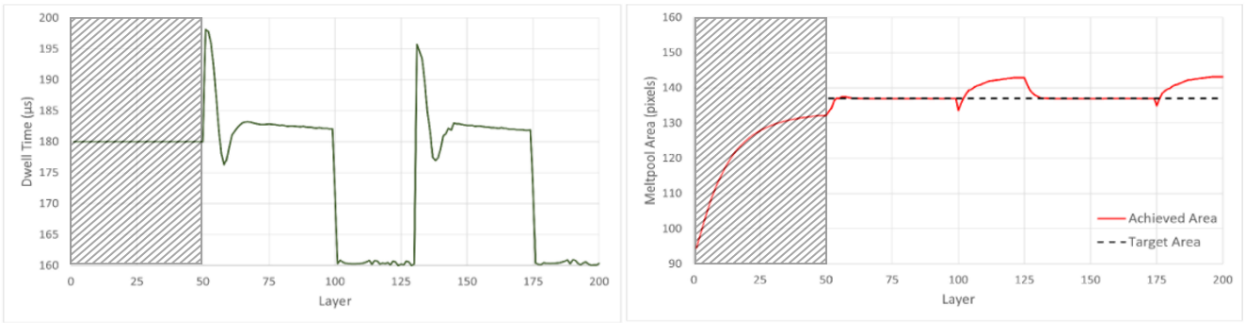

Fig. 5.18 Resulting policy of the SAC agent with fixed target, overhang geometry.

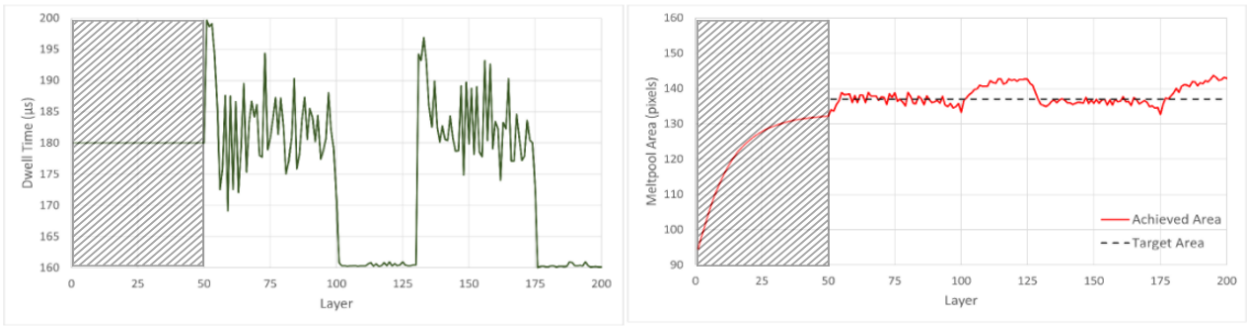

Fig. 5.19 Resulting policy of the SAC-noisy agent with fixed target, overhang geometry.

Table 5.7 Fixed target results comparison of SAC and SAC-noisy, overhang geometry

Controller	Mean Area (pixels)	MAE (pixels)	Ov. Mean Area (pixels)	Ov. MAE (pixels)
Openloop	137	4.60	141.96	5.41
Openloop-noisy	137	4.73	141.98	5.44
SAC	138.09	1.71	140.44	4.12
SAC-noisy	138.17	1.99	140.97	4.18

For consistency with the cuboid case study, and in order to further test the agents, the control problem is now altered by introducing a varying A_{target}. Once again, in order to closely mimic the real process, a noise signal of 1% is also introduced in the observed area signal. The RL controller framework is designed as shown in Table 5.1.

Figure 5.20 depicts the best training process of the SAC agent without noise in the area signal, and with 1% noise in the area signal, after 10 training runs for each case. It is observed that the SAC agent outperforms the SAC-noisy agent, as expected. It achieves a maximum reward of 109.61 out of 150 (maximum theoretically possible), hence approximately 73% training performance. The training metrics are shown in Table 5.8.

Similarly to the previous fixed target overhang case study, it is argued that the agents are significantly challenged with 73% training performance, compared to the correspondent cuboid case study with training performance higher than 95%. For more clarity, the focus of the analysis turns to the resulting policy in order to understand better why the agents are challenged.

Fig. 5.20 Best training curves of the SAC and the SAC-noisy agents. Varying target for the average layer area, overhang geometry.

Table 5.8 Varying target training comparison between SAC and SAC-noisy, overhang geometry

RL agents	Mean Reward	Maximum Reward
SAC	93.60	109.61
SAC-noisy	93.56	107.11

The resulting policy and the achieved average meltpool area in each layer are shown in Figures 5.21 and 5.22. In the bulk regions of the part, it is observed that both agents demonstrate very good performance, managing to maintain the average layer area in the desired value. In the first overhang balcony region, the target area is consistently below the openloop resulting area, hence the agents attempt to minimise the dwell time value to 160μs in order to compensate for the overhang overheating phenomenon. However, the overhang overheating phenomenon is not sufficiently addressed. In the second overhang balcony region, the target area profile resembles the openloop resulting area profile, hence, despite the overhang overheating phenomenon, the agents achieve the desired target in a more satisfactory manner. In all, the SAC controller achieves a MAE of 1.72 pixels and the SAC-noisy controller achieves a MAE of 1.93 pixels.

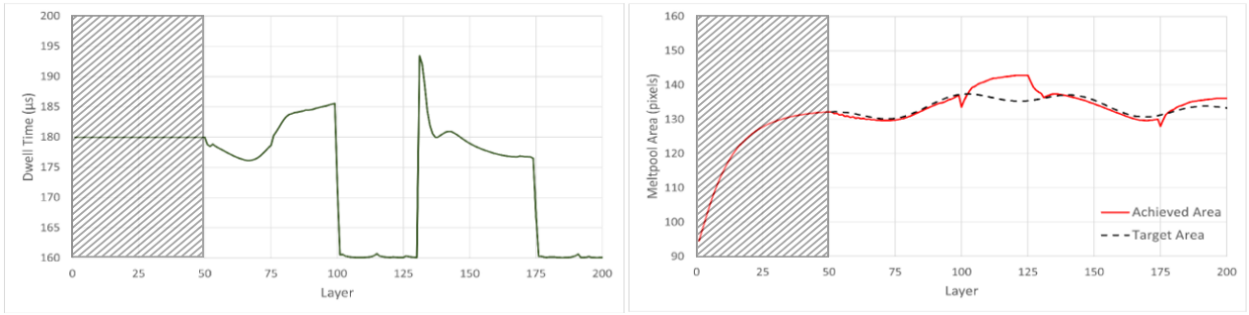

Fig. 5.21 Resulting policy of the SAC agent with varying target, overhang geometry.

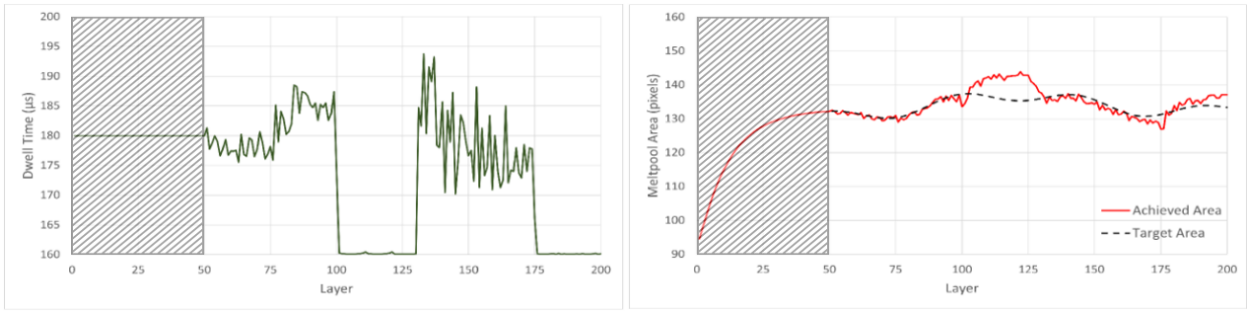

Fig. 5.22 Resulting policy of the SAC-noisy agent with varying target, overhang geometry.

5.5.3 Constrained SAC control

Finally, in this section, RL process control with constraint handling is implemented, as described in the control formulation section. A constraint of $\pm 2\mu$s of the previous action is applied on the action to be taken, and the effective K region is (0, 5). The K values close to 0 saturate the attempted action to the value of the previous action taken, while the K values close to 5 saturate the attempted action to its feasible constraint boundaries. The rest of the K values within the effective K region can provide a more flexible mapping of the attempted action into the feasible constrained action region.

In order to demonstrate RL process control which is safe for real-world deployment, the subsequent intuition is followed. The fixed meltpool area target is the most common target in real-world EBM, since consistent, homogeneously dense parts are the most commercially popular. Moreover, for realistic representation, noise (1%) in the observation signal has to be taken into consideration. Finally, not only the machine's overall limits have to be respected, but also the defined dynamic constraints ($\pm 2\mu$s) have to be satisfied under zero constraint violation. Hence, in this section, the RL agents are trained under these conditions, resulting to RL agents that are safe for deployment. More specifically, the focus is on applying constrained RL control on the cuboid geometry, with fixed target and noisy area signal, and on the overhang geometry, with fixed target and noisy area signal. The correspondent results of SAC-noisy control without constraints have been previously shown in Figures 5.13 and 5.19. In this section, the constrained controller is referred to as CSAC-noisy and it is designed as shown in Table 5.2.

Figure 5.23 depicts the best training process of the CSAC-noisy agent, after 10 training runs for the cuboid geometry. It is observed that the CSAC-noisy agent achieves a mean reward of 137.39 and a maximum reward

of 141.81 out of 150 (maximum theoretically possible), hence approximately 94% training performance. On the other hand, the SAC-noisy agent from the previous case study (no constraints) achieves a mean reward of 139.35 and a maximum reward of 141.91, see Table 5.3. Overall, a deterioration in the training performance is observed because of the constraint introduction, however, it is not significant enough to reflect on the quality of the control performance.

Fig. 5.23 Best training curve of the CSAC-noisy agent. Fixed target for the average layer area, cuboid geometry.

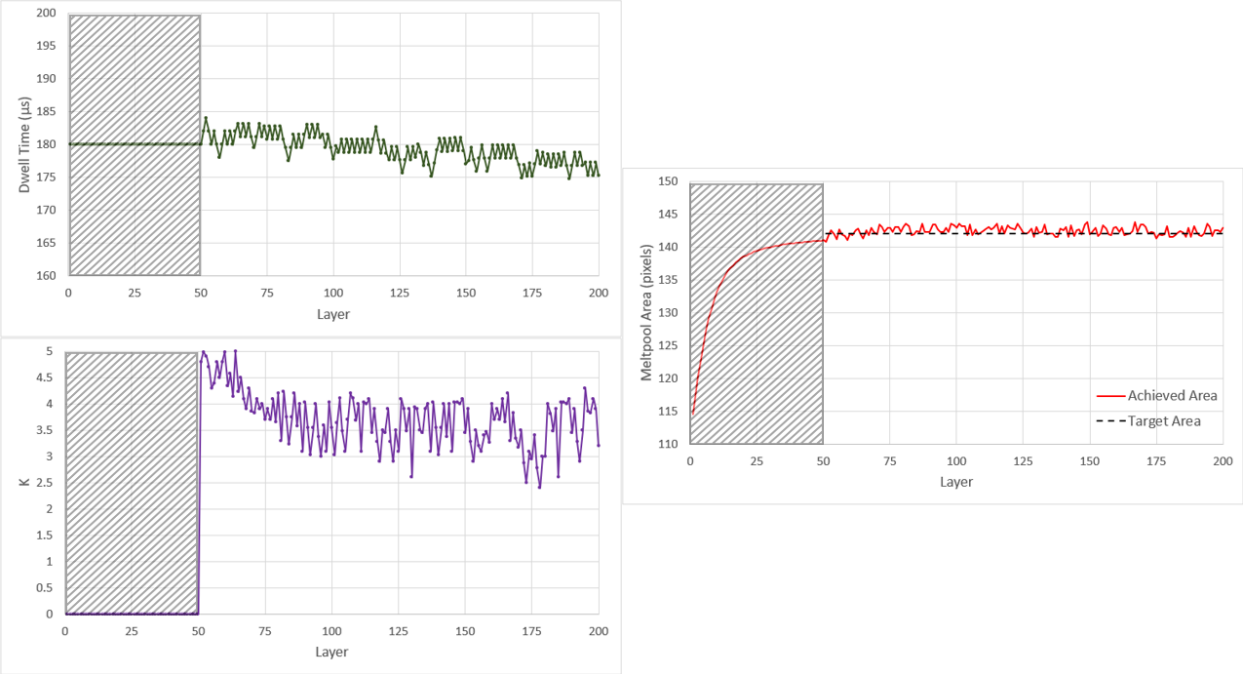

Fig. 5.24 Resulting policy of the CSAC-noisy agent with fixed target, cuboid geometry. Graphs on the left demonstrating the chosen K value and the resulting dwell time value applied.

The resulting policy and the achieved average meltpool area in each layer are shown in Figure 5.24. It is observed that the CSAC-noisy agent demonstrates very good performance, managing to maintain the average layer area in a mean of 142.46 pixels, with a MAE of 0.63 pixels. In the first layers, it is observed that the

CSAC-noisy agent chooses high K values, in order to apply the largest dwell time changes possible, and reach the desired area target quickly. Then, the K values drop, but remain in high levels, allowing for more flexible mapping of the attempted dwell time actions into the feasible region, but still applying large dwell time changes when necessary. In general, it is observed that the CSAC-noisy agent attempts to follow the same policy as the one observed in Figure 5.13. Despite the introduction of constraints, the agent still manages to capture the dynamic behaviour of the control environment and attempts to follow a good policy, while maintaining a zero constraint violation framework. The constraint satisfaction indeed poses challenges, e.g., the necessary steep actions are delayed, but the control performance is similar to the one observed in Figure 5.13, due to the simplicity of the cuboid geometry dynamics. In conclusion, it is argued that, despite the constraint introduction, the heat accumulation issues observed in the corresponding openloop results of Figure 5.9 are sufficiently addressed.

Figure 5.25 depicts the best training process of the CSAC-noisy agent, after 10 training runs for the overhang geometry. It is observed that the CSAC-noisy agent achieves a mean reward of 96.46 and a maximum reward of 107.91 out of 150 (maximum theoretically possible), hence approximately 72% training performance. On the other hand, the SAC-noisy agent from the previous case study (no constraints) achieves a mean reward of 98.44 and a maximum reward of 110.23, see Table 5.6. Overall, a deterioration in the training performance is observed because of the constraint introduction, which seems significant enough to reflect on the quality of the control performance.

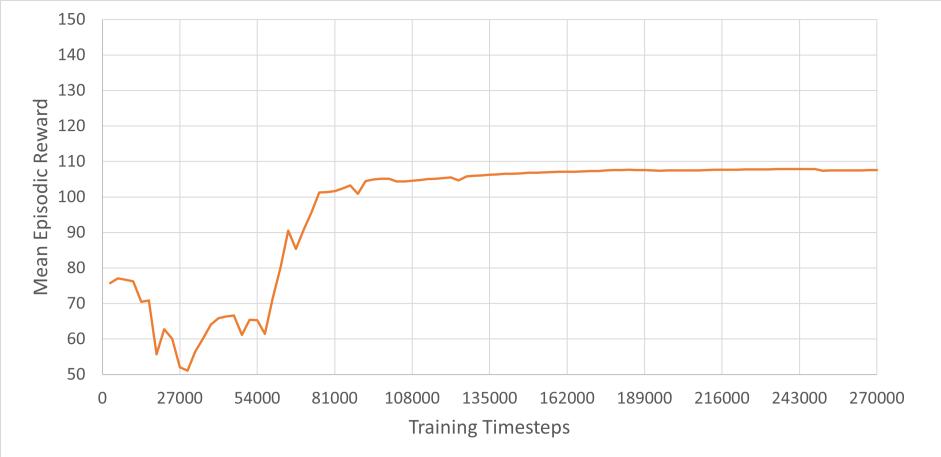

Fig. 5.25 Best training curve of the CSAC-noisy agent. Fixed target for the average layer area, overhang geometry.

The resulting policy and the achieved average meltpool area in each layer are shown in Figure 5.26. It is observed that the CSAC-noisy agent does not demonstrate good performance, maintaining the average layer area in a mean of 138.42 pixels, with a MAE of 2.91 pixels. In the first layers, it is observed that the CSAC-noisy agent chooses high K values, in order to apply the largest dwell time changes possible, and reach the desired area target quickly. Then, the K values drop, allowing for more flexible mapping of the attempted dwell time actions into the feasible region, but remain in relatively high levels, still applying large dwell time changes when necessary. In the first overhang balcony region, the agent again chooses high K values, in order to apply the largest dwell time changes possible, and minimise the dwell time quickly to compensate for the overhang overheating phenomenon. Once the dwell time is minimised, the agent chooses very low K values

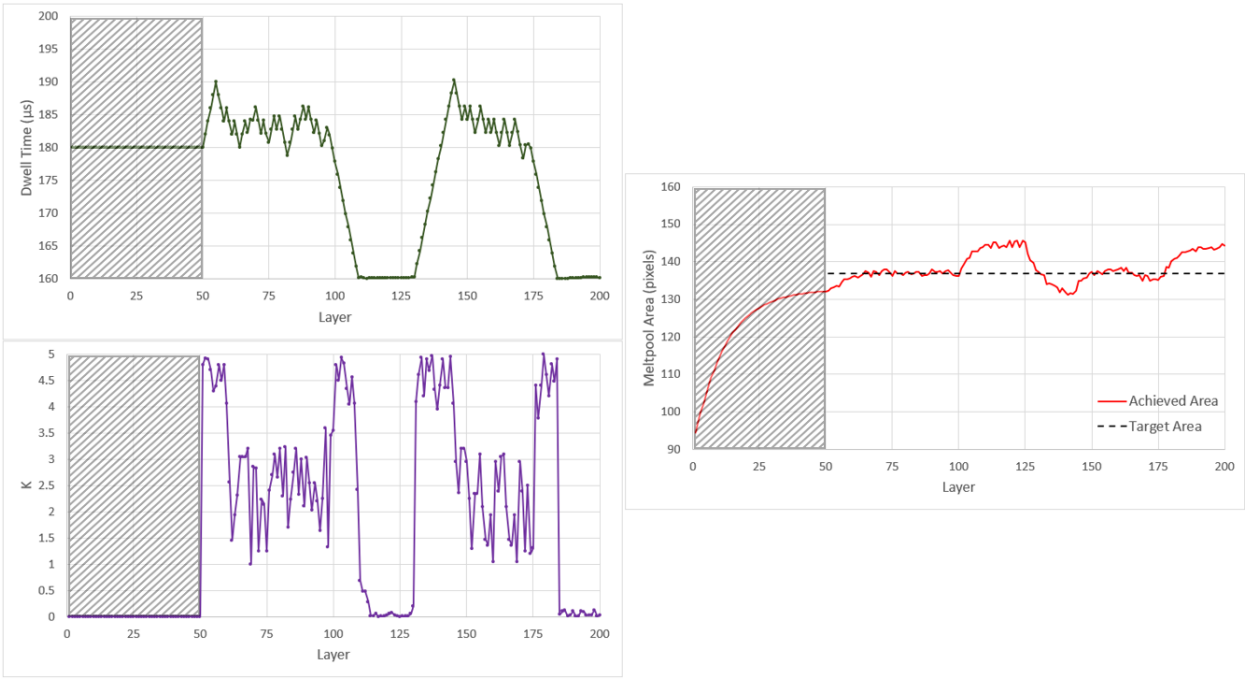

Fig. 5.26 Resulting policy of the CSAC-noisy agent with fixed target, overhang geometry. Graphs on the left demonstrating the chosen K value and the resulting dwell time value applied.

in order to maintain the dwell time to the same, minimum value. After the overhang balcony region, the agent again chooses high K values, and the same pattern is repeated. In general, it is observed that the CSAC-noisy agent attempts to follow the same policy as the one observed in Figure 5.19. Despite the introduction of constraints, the agent still manages to capture the dynamic behaviour of the control environment and attempts to follow a good policy, while maintaining a zero constraint violation framework. However, due to the constraint satisfaction, the necessary steep actions are delayed, and the control performance is inferior to the one observed in Figure 5.19, due to the steep dwell time change needs in the overhang geometry. In conclusion, it is argued that, despite the constraint introduction, the agent attempts to follow a reasonable policy, but the heat accumulation issues observed in the corresponding openloop results of Figure 5.9 are not sufficiently addressed.

5.6 Discussion and future work

This work presents a RL framework for process control in EBM, using real-world based simulation models for the 3D build process. The key novelty, from a methodology perspective, lies in the implementation of auto-tuned, constraint-aware RL controllers, tailored to maintain the desired meltpool area target, which is a key variable affecting part quality. From an application perspective, the main contribution lies in the development of real-world based simulation platforms capable of representing both simple and complex (overhang) EBM geometries. Emphasis is placed both on control performance, and on safety settings, laying the foundation for safe, real-world deployment of RL controllers in EBM.

The proposed RL framework is validated through a series of control problems for two primary geometries: cuboid and overhang structures. For the cuboid geometry, the SAC agents demonstrate exceptional control

capabilities, achieving MAEs as low as 0.52 pixels under fixed target noisy conditions and 0.65 pixels under varying target noisy conditions. These results indicate the ability of RL to effectively maintain the desired meltpool area target, even in the presence of observational noise. In contrast, the overhang geometry posed a more significant challenge due to its inherently complex thermal dynamics and localised overheating phenomena. While the SAC agent shows reasonable performance in the bulk regions of the geometry, it struggles to sufficiently mitigate overheating in the balcony regions. This underperformance stems from the need for more aggressive or anticipatory control strategies, which the current action space, limited to a narrow dwell time window, may not fully support.

The integration of constraints in the RL framework, reflecting practical limitations such as bounded action change rates ($\pm 2\mu$s), introduces a critical tradeoff. While the constrained SAC agent under noise (CSAC-noisy) successfully demonstrates safe behaviour with zero constraint violation, it also exhibits a reduction in control performance, particularly in the overhang case. This illustrates the inherent balance between enforcing safety and maintaining high control performance. Nevertheless, the CSAC-noisy agent retains the capacity to generalise effective policies within safe bounds, proving its capability for deployment in real-world EBM applications. A key example is the agent's action behaviour in Figure 5.13, in which a spike in the dwell time is observed. This spike, in a real-world setting, is either infeasible or can be damaging to the part's quality. Hence, the constraint integration in the RL framework, see Figure 5.24, successfully addresses this safety-critical challenge.

For future research endeavours, we encourage work towards multi-variable RL process control, e.g., dwell time, beam current and scanning strategy. We argue that this would provide additional degrees of freedom in the control problem, enabling more precise and responsive control, especially in challenging geometries such as overhang structures. Moreover, we find that a comparison of the RL agents in this work with other established control techniques such as model predictive control frameworks can prove substantial for further intuition and challenge identification. Finally, it is in our future endeavours to implement the proposed, safe for real-world deployment, RL framework, into a real-world EBM machine, in order to validate all the simulation results presented in this work.

5.7 Declarations

5.7.1 Funding

This research work was funded by the UK Engineering and Physical Sciences Research Council (EP/P006566/1) and iCASE (EP/T517835/1), with contributions from Wayland Additive Ltd.

5.8 Compliance with Ethical Standards

The authors have no further ethical issues to disclose.

Chapter 6

Discussion and future research directions

This thesis presents a substantial advancement in the development and application of RL for process control in PBF. Building upon a foundation of theoretical insights and practical implementations, this research work introduces a class of intelligent, RL control frameworks, validated across both SLM and EBM processes. Key contributions include the formulation of stable RL algorithms and constrained RL frameworks, the development of realistic and computationally efficient 3D simulation platforms for multi-layer part fabrication, and comprehensive evaluation across a range of geometrical and thermal scenarios. These efforts collectively contribute to the vision of trustworthy manufacturing systems, capable of operating under complex, uncertain, and safety-critical conditions.

6.1 Contributions

The first publication chapter (Chapter 2) of this thesis focuses on the critical aspect of stability in RL control for PBF. It provides a comprehensive investigation into the challenges of ensuring stable learning in RL, particularly under the complex, nonlinear dynamics of PBF processes. It concludes by acknowledging the importance of Lyapunov stability insight in stable RL frameworks, encouraging more research towards Lyapunov-based stability in RL.

Following the above intuition, the second publication chapter (Chapter 3) explores the stability literature in a more practical perspective, with stronger critique on Lyapunov RL methods. More specifically, it critiques existing Lyapunov-based RL frameworks, based on their dependence on restrictive assumptions and convergence guarantees, which are often impractical in real-world applications with unknown dynamics. Hence, building upon this foundation, the second publication chapter (Chapter 3) introduces a novel RL algorithm, which enhances stability without imposing strict theoretical constraints, hence without compromising RL's inherent flexibility. Inspired by Lyapunov principles, but designed with practical adaptability in mind, this algorithm includes an auto-tuned penalty mechanism integrated into the existing RL framework, which penalises undesirable behaviours. This method is validated using a newly developed 3D SLM simulation platform capable of representing full multi-layer builds, making this the first known demonstration of RL control applied to complete 3D PBF parts. The results show clear improvements over PID control and highlight the proposed method's stability and performance benefits.

The third publication chapter (Chapter 4) shifts focus to the critical challenge of constraint satisfaction in RL control systems. It introduces a constrained RL framework which explicitly incorporates operational boundaries into the learning process. This method allows the agent to learn in safe action regions, while it also includes a scalar factor to adjust constraint intensity. While effective, with guaranteed zero constraint violation, the control performance is sensitive to the choice of the intensity factor, requiring careful manual tuning and limiting adaptability under changing conditions.

To address this limitation, the fourth publication chapter (Chapter 5) proposes an auto-tuned version of this constraint-aware RL framework, which dynamically tunes constraint intensity during both training and deployment. This removes the need for fixed hyperparameter tuning and enhances the practicality of RL in real-world manufacturing settings. This method is tested in a newly developed 3D EBM simulation platform, based on real-world data, making this the first known demonstration of RL control applied to real-world based models, including cuboid and overhang strucutres. The results demonstrate that in most cases, RL agents can achieve safe, effective control even under observational noise and strict action constraints. Importantly, the proposed constrained RL method offers a guaranteed zero constraint violation framework, paving the way for safer deployment in industrial scenarios.

6.2 Remaining challenges and future work

Despite the aforementioned progress made, there are still challenges which are recognised as potential barriers for RL adaption in PBF manufacturing systems. A remaining key challenge is the inherent unpredictability of RL training outcomes. As demonstrated in this research work, while successful policies can be learned, there is no general guarantee of RL convergence, especially in highly uncertain or partially observable environments.

Importantly, the above challenge does not render RL unsuitable for such applications, e.g., PBF. On the contrary, one of the key findings of this thesis is that the integration of explicit safety constraints into the RL framework offers a powerful method to address this concern. By enforcing action boundaries and constraint satisfaction during both training and deployment, constrained RL approaches can ensure that even if the agent's performance fluctuates, its behaviour remains within safe operational limits. This is especially important in safety-critical industrial settings, in which safety often outweighs absolute control optimality. As previously mentioned in this thesis, the following tradeoff holds true. While constraint-aware RL may sacrifice some degree of control performance, it offers critical assurances that unsafe behaviours are avoided, which is a non-negotiable requirement for real-world adoption. Nevertheless, it is argued that addressing the unpredictability challenge in RL, would make RL control methods more reliable in terms of expected control performance.

Regarding RL, another important aspect is how the hyperparameters are chosen for each task. The choice of the hyperparameters requires experience and good knowledge of the literature. For instance, in this research work, the discount factor value is chosen as the default value used overwhelmingly in the literature, whereas other hyperparameters, such as learning rate, are chosen after experimental investigation. It is argued that a thorough analysis and comparison study on how different hyperparameters can result to different RL control performance can be of high importance, especially in still poorly explored (for control) areas such as PBF.

Another key challenge is noted regarding the developed models in this thesis, since the developed models have limited control capability. They provide a high-level layer-by-layer access, hence layer-by-layer control

capability, while control is arguably also needed within a low-level setting, within the layer built. As a result, there is a need to also investigate low-level control schemes, such as track-by-track and point-by-point approaches. Regarding the control variables, future work should also explore multi-variable control techniques, such as simultaneously controlling dwell time, beam power, and scanning strategy. This would provide greater flexibility and precision, particularly for complex part geometries such as overhangs, in which localised thermal management is crucial.

Moreover, although the developed models comprise a great improvement compared to most of the existing models found in the literature, they still represent idealised models. Real-world PBF processes involve stochastic behaviours and disturbances that are not fully captured in simulation. Hence, future work should include validating the proposed RL frameworks on actual PBF machines, which will also help to identify gaps between simulated and real-world performance.

In summary, this thesis makes substantial progress in advancing safe and effective RL process control in PBF. It is acknowledged that the aforementioned remaining challenges should be addressed for more reliable RL frameworks and deeper control intuition. However, it is argued that the real-world validation remains the critical next step. Future work must focus on deploying these methods on actual PBF machines to test their robustness against real process conditions. This is an essential step for establishing RL as a trustworthy control solution in industrial, real-world PBF settings.

References

[1] Ana Vafadar, Ferdinando Guzzomi, Alexander Rassau, and Kevin Hayward. Advances in metal additive manufacturing: A review of common processes, industrial applications, and current challenges. *Applied Sciences*, 11(3), 2021. ISSN 2076-3417. doi: 10.3390/app11031213. URL https://www.mdpi.com/2076-3417/11/3/1213.

[2] Sohini Chowdhury, N. Yadaiah, Chander Prakash, Seeram Ramakrishna, Saurav Dixit, Lovi Raj Gupta, and Dharam Buddhi. Laser powder bed fusion: a state-of-the-art review of the technology, materials, properties & defects, and numerical modelling. *Journal of Materials Research and Technology*, 20: 2109–2172, 2022. ISSN 2238-7854. doi: https://doi.org/10.1016/j.jmrt.2022.07.121. URL https://www.sciencedirect.com/science/article/pii/S2238785422011607.

[3] Naol Dessalegn Dejene and Hirpa G. Lemu. Current status and challenges of powder bed fusion-based metal additive manufacturing: Literature review. *Metals*, 13(2), 2023. ISSN 2075-4701. doi: 10.3390/met13020424. URL https://www.mdpi.com/2075-4701/13/2/424.

[4] Shawn Moylan, Eric Whitenton, Brandon Lane, and John Slotwinski. Infrared thermography for laser-based powder bed fusion additive manufacturing processes. *AIP Conference Proceedings*, 1581(1): 1191–1196, 2014. doi: 10.1063/1.4864956. URL https://aip.scitation.org/doi/abs/10.1063/1.4864956.

[5] Gustavo Tapia, Saad Khairallah, Manyalibo Matthews, Wayne E. King, and Alaa Elwany. Gaussian process-based surrogate modeling framework for process planning in laser powder-bed fusion additive manufacturing of 316l stainless steel. *The International Journal of Advanced Manufacturing Technology*, 94(9):3591–3603, 2018. ISSN 1433-3015. doi: 10.1007/s00170-017-1045-z. URL https://doi.org/10.1007/s00170-017-1045-z.

[6] Changchun Zhang, Tingting Liu, Wenhe Liao, Huiliang Wei, and Ling Zhang. Investigation of the laser powder bed fusion process of ti-6.5al-3.5mo-1.5zr-0.3si alloy. *Chinese Journal of Mechanical Engineering*, 36:32, 03 2023. doi: 10.1186/s10033-023-00863-z.

[7] W. E. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, S. A. Khairallah, and A. M. Rubenchik. Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges. *Applied Physics Reviews*, 2(4):041304, 2015. doi: 10.1063/1.4937809. URL https://doi.org/10.1063/1.4937809.

[8] William E. Frazier. Metal additive manufacturing: A review. *Journal of Materials Engineering and Performance*, 23(6):1917–1928, 2014. ISSN 1544-1024. URL https://doi.org/10.1007/s11665-014-0958-z.

[9] Jorrit Voigt, Thomas Bock, Uwe Hilpert, Ralf Hellmann, and Michael Moeckel. Increased relative density and characteristic melt pool signals at the edge in pbf-lb/m. *Additive Manufacturing*, 57: 102798, 2022. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2022.102798. URL https://www.sciencedirect.com/science/article/pii/S2214860422001993.

[10] Rob Snell, Sam Tammas-Williams, Lova Chechik, Alistair Lyle, Everth Hernández-Nava, Charlotte Boig, George Panoutsos, and I. Todd. Methods for rapid pore classification in metal additive manufacturing. *JOM*, 72:1–9, 09 2019. doi: 10.1007/s11837-019-03761-9.

[11] Shunyu Liu and Yung C. Shin. Additive manufacturing of ti6al4v alloy: A review. *Materials & Design*, 164:107552, 2019. ISSN 0264-1275. doi: https://doi.org/10.1016/j.matdes.2018.107552. URL https://www.sciencedirect.com/science/article/pii/S026412751830916X.

[12] Jean-Pierre Kruth, Joost Duflou, Peter Mercelis, Jonas Van Vaerenbergh, Tom Craeghs, and Johan De Keuster. On-line monitoring and process control in selective laser melting and laser cutting. *Proceedings of the 5th Lane Conference, Laser Assisted Net Shape Engineering*, 1(1):23–37, 2007.

[13] W. Sames, Franziska List, Sreekanth Pannala, Ryan Dehoff, and Sudarsanam Babu. The metallurgy and processing science of metal additive manufacturing. *International Materials Reviews*, 61:1–46, 03 2016. doi: 10.1080/09506608.2015.1116649.

[14] Francis Ogoke and Amir Barati Farimani. Thermal control of laser powder bed fusion using deep reinforcement learning. *Additive Manufacturing*, 46:102033, 2021. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2021.102033. URL https://www.sciencedirect.com/science/article/pii/S2214860421001986.

[15] M. Matsumoto, M. Shiomi, K. Osakada, and F. Abe. Finite element analysis of single layer forming on metallic powder bed in rapid prototyping by selective laser processing. *International Journal of Machine Tools and Manufacture*, 42:61–67, 01 2002. doi: 10.1016/S0890-6955(01)00093-1.

[16] M. Zäh and S. Lutzmann. Modelling and simulation of electron beam melting. *Production Engineering*, 4:15–23, 02 2010. doi: 10.1007/s11740-009-0197-6.

[17] Mustafa Megahed, Hans-Wilfried Mindt, Narcisse N'Dri, Hongzhi Duan, and Olivier Desmaison. Metal additive-manufacturing process and residual stress modeling. *Integrating Materials and Manufacturing Innovation*, 5, 02 2016. doi: 10.1186/s40192-016-0047-2.

[18] Qian Wang, Panagiotis (Pan) Michaleris, Abdalla R. Nassar, Jeffrey E. Irwin, Yong Ren, and Christopher B. Stutzman. Model-based feedforward control of laser powder bed fusion additive manufacturing. *Additive Manufacturing*, 31, January 2020. ISSN 2214-8604. doi: 10.1016/j.addma.2019.100985.

[19] Rongpei Shi, Saad Khairallah, Tae Wook Heo, Matthew Rolchigo, Joseph McKeown, and Manyalibo Matthews. Integrated simulation framework for additively manufactured ti-6al-4v: Melt pool dynamics, microstructure, solid-state phase transformation, and microelastic response. *JOM*, 71:1–16, 06 2019. doi: 10.1007/s11837-019-03618-1.

[20] Wentao Yan, F Lin, and W Liu. An effective finite element heat transfer model for electron beam melting process. In *Proceedings of the Advances in Materials and Processing Technologies Conference, Madrid, Spain*, pages 14–17, 2015.

[21] I.A. Roberts, Chang Wang, R. Esterlein, M. Stanford, and Diane Mynors. A three-dimensional finite element analysis of the temperature field during laser melting of metal powders in additive layer manufacturing. *International Journal of Machine Tools and Manufacture*, 49:916–923, 10 2009. doi: 10.1016/j.ijmachtools.2009.07.004.

[22] Manuela Galati, Oscar Di Mauro, and Luca Iuliano. Finite element simulation of multilayer electron beam melting for the improvement of build quality. *Crystals*, 10(6), 2020. ISSN 2073-4352. doi: 10.3390/cryst10060532. URL https://www.mdpi.com/2073-4352/10/6/532.

[23] T W Eagar and N S Tsai. Temperature fields produced by traveling distributed heat sources. *Weld. Res. Suppl.; (United States)*, 12 1983. URL https://www.osti.gov/biblio/5782268.

[24] Alexander J. Wolfer, Jeremy Aires, Kevin Wheeler, Jean-Pierre Delplanque, Alexander Rubenchik, Andy Anderson, and Saad Khairallah. Fast solution strategy for transient heat conduction for arbitrary scan paths in additive manufacturing. *Additive Manufacturing*, 30:100898, 2019. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2019.100898. URL https://www.sciencedirect.com/science/article/pii/S2214860419303446.

[25] William Mycroft, Mordechai Katzman, Sam Tammas-Williams, Everth Hernández-Nava, George Panoutsos, I. Todd, and Visakan Kadirkamanathan. A data-driven approach for predicting printability in metal additive manufacturing processes. *Journal of Intelligent Manufacturing*, 31, 10 2020. doi: 10.1007/s10845-020-01541-w.

[26] Nadia Kouraytem, Xuxiao Li, Wenda Tan, Branden Kappes, and Ashley D Spear. Modeling process-structure-property relationships in metal additive manufacturing: a review on physics-driven versus data-driven approaches. *Journal of Physics: Materials*, 4(3):032002, 04 2021. doi: 10.1088/2515-7639/abca7b. URL https://doi.org/10.1088/2515-7639/abca7b.

[27] Scott C. Jensen, Jay D. Carroll, Priya R. Pathare, David J. Saiz, Jonathan W. Pegues, Brad L. Boyce, Bradley H. Jared, and Michael J. Heiden. Long-term process stability in additive manufacturing. *Additive Manufacturing*, 61:103284, 2023. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2022.103284. URL https://www.sciencedirect.com/science/article/pii/S221486042200673X.

[28] Volker Renken, Axel von Freyberg, Kevin Schünemann, Felix Pastors, and Andreas Fischer. In-process closed-loop control for stabilising the melt pool temperature in selective laser melting. *Progress in Additive Manufacturing*, 4, 12 2019. doi: 10.1007/s40964-019-00083-9.

[29] Jorge Mireles, Cesar Terrazas, Sara Gaytan, David Roberson, and Ryan Wicker. Closed-loop automatic feedback control in electron beam melting. *The International Journal of Advanced Manufacturing Technology*, 78, 05 2015. doi: 10.1007/s00170-014-6708-4.

[30] J. P. Kruth, Peter Mercelis, Jonas Van Vaerenbergh, and Tom Craeghs. Feedback control of selective laser melting. In *Proceedings of the 15th International Symposium on Electromachining*, pages 421–426, 2007. URL https://api.semanticscholar.org/CorpusID:137846354.

[31] Dominic Liao-McPherson, Efe C. Balta, Ryan Wuest, Alisa Rupenyan, and John Lygeros. In-layer thermal control of a multi-layer selective laser melting process. In *2022 European Control Conference (ECC)*, pages 1678–1683, 2022. doi: 10.23919/ECC55457.2022.9838031.

[32] Ema Vasileska, Ali Gökhan Demir, Bianca Maria Colosimo, and Barbara Previtali. Layer-wise control of selective laser melting by means of inline melt pool area measurements. *Journal of Laser Applications*, 32(2):022057, 05 2020. ISSN 1042-346X. doi: 10.2351/7.0000108. URL https://doi.org/10.2351/7.0000108.

[33] Tianyu Jiang, Mengying Leng, and Xu Chen. Control-oriented mechatronic design and data analytics for quality-assured laser powder bed fusion additive manufacturing. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1319–1324, 07 2021. doi: 10.1109/AIM46487.2021.9517393.

[34] Zhimin Xi. Model predictive control of melt pool size for the laser powder bed fusion process under process uncertainty. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 8, 07 2021. doi: 10.1115/1.4051746.

[35] Barış Kavas, Efe Balta, Michael Tucker, Alisa Rupenyan-Vasileva, John Lygeros, and Markus Bambach. Layer-to-layer closed-loop feedback control application for inter-layer temperature stabilization in laser powder bed fusion. *Additive Manufacturing*, 2023. URL https://doi.org/10.1016/j.addma.2023.103847.

[36] Ahmed M. Faizan Mohamed, Francesco Careri, Raja H.U. Khan, Moataz M. Attallah, and Leonardo Stella. A novel porosity prediction framework based on reinforcement learning for process parameter optimization in additive manufacturing. *Scripta Materialia*, 255:116377, 2025. ISSN 1359-6462. doi: https://doi.org/10.1016/j.scriptamat.2024.116377. URL https://www.sciencedirect.com/science/article/pii/S1359646224004123.

[37] Yingjie Zhang and Wentao Yan. Applications of machine learning in metal powder-bed fusion in-process monitoring and control: status and challenges. *Journal of Intelligent Manufacturing*, 34, 08 2023. doi: 10.1007/s10845-022-01972-7.

[38] Francesco Lupi, Alessio Pacini, and Michele Lanzetta. Laser powder bed additive manufacturing: A review on the four drivers for an online control. *Journal of Manufacturing Processes*, 103:413–429, 2023. ISSN 1526-6125. doi: https://doi.org/10.1016/j.jmapro.2023.08.022. URL https://www.sciencedirect.com/science/article/pii/S152661252300796X.

[39] Rui Nian, Jinfeng Liu, and Biao Huang. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139:106886, 2020. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2020.106886. URL https://www.sciencedirect.com/science/article/pii/S0098135420300557.

[40] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. A Bradford book. MIT Press, 1998. ISBN 9780262193986. URL https://books.google.co.uk/books?id=CAFR6IBF4xYC.

[41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 07 2017. doi: 10.48550/arXiv.1707.06347.

[42] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[43] Fabio Pardo. Tonic: A deep reinforcement learning library for fast prototyping and benchmarking. *ArXiv*, abs/2011.07537, 2020.

[44] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[45] Wenqi Cui, Yan Jiang, and Baosen Zhang. Reinforcement learning for optimal primary frequency control: A lyapunov approach. *IEEE Transactions on Power Systems*, pages 1–1, 2022. doi: 10.1109/TPWRS.2022.3176525.

[46] Wenqi Cui and Baosen Zhang. Lyapunov-regularized reinforcement learning for power system transient stability. *IEEE Control Systems Letters*, 6:974–979, 2022. doi: 10.1109/LCSYS.2021.3088068.

[47] Yuanyuan Shi, Guannan Qu, Steven Low, Anima Anandkumar, and Adam Wierman. Stability constrained reinforcement learning for real-time voltage control. In *2022 Annual American Control Conference (ACC)*. IEEE, 2022.

[48] Zhenyi Yuan, Changhong Zhao, and Jorge Cortés. Reinforcement learning for distributed transient frequency control with stability and safety guarantees. *Systems & Control Letters*, 185:105753, 2024. ISSN 0167-6911. doi: https://doi.org/10.1016/j.sysconle.2024.105753. URL https://www.sciencedirect.com/science/article/pii/S0167691124000410.

[49] Minghao Han, Lixian Zhang, Jun Wang, and Wei Pan. Actor-critic reinforcement learning for control with stability guarantee. *IEEE Robotics and Automation Letters*, PP:1–1, 07 2020. doi: 10.1109/LRA.2020.3011351.

[50] Lixian Zhang, Ruixian Zhang, Tong Wu, Rui Weng, Minghao Han, and Ye Zhao. Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5435–5444, 12 2021. doi: 10.1109/TNNLS.2021.3084685.

[51] Eitan Altman. *Constrained Markov Decision Processes*. Routledge, 1999. URL https://api.semanticscholar.org/CorpusID:14906227.

[52] Harshit Sikchi, Wenxuan Zhou, and David Held. Lyapunov barrier policy optimization. *arXiv preprint arXiv:2103.09230*, 2021.

[53] Kazumi Kasaura, Shuwa Miura, Tadashi Kozuno, Ryo Yonetani, Kenta Hoshino, and Yohei Hosoe. Benchmarking actor-critic deep reinforcement learning algorithms for robotics control with action constraints. *IEEE Robotics and Automation Letters*, 8(8):4449–4456, 2023. doi: 10.1109/LRA.2023.3284378.

[54] Lixian Zhang, Ruixian Zhang, Tong Wu, Rui Weng, Minghao Han, and Ye Zhao. Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5435–5444, 12 2021. doi: 10.1109/TNNLS.2021.3084685.

[55] Taha Al-Saadi, J. Anthony Rossiter, and George Panoutsos. Initial investigation of online control system for selective laser melting process: Multi-layer level. In *2024 UKACC 14th International Conference on Control (CONTROL)*, pages 268–273, 2024. doi: 10.1109/CONTROL60310.2024.10532025.

[56] N.S. Johnson, P.S. Vulimiri, A.C. To, X. Zhang, C.A. Brice, B.B. Kappes, and A.P. Stebner. Invited review: Machine learning for materials developments in metals additive manufacturing. *Additive Manufacturing*, 36:101641, 2020. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2020.101641.

[57] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, 12 2017.

[58] Michal Piovarči, Michael Foshey, Jie Xu, Timmothy Erps, Vahid Babaei, Piotr Didyk, Szymon Rusinkiewicz, Wojciech Matusik, and Bernd Bickel. Closed-loop control of direct ink writing via reinforcement learning. *ACM Transactions on Graphics*, 41:1–10, 07 2022. doi: 10.1145/3528223.3530144.

[59] Audelia G. Dharmawan, Yi Xiong, Shaohui Foong, and Gim Song Soh. A model-based reinforcement learning and correction framework for process control of robotic wire arc additive manufacturing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4030–4036, 2020. doi: 10.1109/ICRA40945.2020.9197222.

[60] Pierre-Alexandre Bliman. A convex approach to robust stability for linear systems with uncertain scalar parameters. *SIAM Journal on Control and Optimization*, 42(6):2016–2042, 2004. doi: 10.1137/S0363012901398691.

[61] Francesco Amato, Giuseppe Carannante, Gianmaria De Tommasi, and Alfredo Pironti. Input–output finite-time stability of linear systems: Necessary and sufficient conditions. *IEEE Transactions on Automatic Control*, 57(12):3051–3063, 2012. doi: 10.1109/TAC.2012.2199151.

[62] Yang Tang, Xiaotai Wu, Peng Shi, and Feng Qian. Input-to-state stability for nonlinear systems with stochastic impulses. *Automatica*, 113:108766, 2020. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2019.108766.

[63] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2018.

[64] Evgenii Nikishin, Pavel Izmailov, Ben Athiwaratkun, Dmitrii Podoprikhin, T. Garipov, Pavel Shvechikov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Improving stability in deep reinforcement learning with weight averaging. In *In Uncertainty in artificial intelligence workshop on uncertainty in Deep learning*, 2018.

[65] Yinlam Chow, Ofir Nachum, Edgar A. Duéñez-Guzmán, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *NeurIPS*, 2018.

[66] Ming Jin and Javad Lavaei. Stability-certified reinforcement learning: A control-theoretic perspective. *IEEE Access*, PP:1–1, 12 2020. doi: 10.1109/ACCESS.2020.3045114.

[67] Maopeng Ran, Juncheng Li, and Lihua Xie. Reinforcement-learning-based disturbance rejection control for uncertain nonlinear systems. *IEEE Transactions on Cybernetics*, PP:1–13, 03 2021. doi: 10.1109/TCYB.2021.3060736.

[68] D. Dev Singh, T. Mahender, and Avala Raji Reddy. Powder bed fusion process: A brief review. *Materials Today: Proceedings*, 46:350–355, 2021. ISSN 2214-7853. doi: https://doi.org/10.1016/j.matpr.2020.08.415. URL https://www.sciencedirect.com/science/article/pii/S2214785320362878. 2nd International Conference on Manufacturing Material Science and Engineering.

[69] Shunyu Liu and Yung C. Shin. Additive manufacturing of ti6al4v alloy: A review. *Materials & Design*, 164:107552, 2019. ISSN 0264-1275. doi: https://doi.org/10.1016/j.matdes.2018.107552. URL https://www.sciencedirect.com/science/article/pii/S026412751830916X.

[70] Runze Huang, Matthew Riddle, Diane Graziano, Joshua Warren, Sujit Das, Sachin Nimbalkar, Joe Cresko, and Eric Masanet. Energy and emissions saving potential of additive manufacturing: the case of lightweight aircraft components. *Journal of Cleaner Production*, 135:1559–1570, 2016. ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2015.04.109. URL https://www.sciencedirect.com/science/article/pii/S0959652615004849.

[71] Stylianos Vagenas and George Panoutsos. Stability in reinforcement learning process control for additive manufacturing. *IFAC-PapersOnLine*, 56(2):4719–4724, 2023. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2023.10.1233. URL https://www.sciencedirect.com/science/article/pii/S2405896323016373. 22nd IFAC World Congress.

[72] Balaji Soundararajan, Daniele Sofia, Diego Barletta, and Massimo Poletto. Review on modeling techniques for powder bed fusion processes based on physical principles. *Additive Manufacturing*, 47:102336, 2021. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2021.102336.

[73] D. Rosenthal. Mathematical theory of heat distribution during welding and cutting. *Welding Journal*, 20:220–234, 1941. URL https://api.semanticscholar.org/CorpusID:210757123.

[74] Pradeep Juneja, Sandeep Sunori, A Sharma, H Pathak, Vikas Joshi, and P Bhasin. A review on control system applications in industrial processes. *IOP Conference Series: Materials Science and Engineering*, 1022:012010, 01 2021. doi: 10.1088/1757-899X/1022/1/012010.

[75] K.J. Åström and T. Hägglund. *Advanced PID Control*. ISA-The Instrumentation, Systems, and Automation Society, 2006. ISBN 9781556179426. URL https://books.google.com.om/books?id=XcseAQAAIAAJ.

[76] Volker Renken, Lutz Lübbert, Hendrik Blom, Axel von Freyberg, and Andreas Fischer. Model assisted closed-loop control strategy for selective laser melting. *Procedia CIRP*, 74:659–663, 2018. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2018.08.053. URL https://www.sciencedirect.com/science/article/pii/S2212827118308370. 10th CIRP Conference on Photonic Technologies [LANE 2018].

[77] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/mniha16.html.

[78] Phani B. Sistu and B. Wayne Bequette. Nonlinear model-predictive control: Closed-loop stability analysis. *AIChE Journal*, 42(12):3388–3402, 1996. doi: https://doi.org/10.1002/aic.690421210. URL https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690421210.

[79] John A Shaw. *The PID control algorithm*, volume 2. Process control solutions, 2003. URL https://www.miataturbo.net/attachments/megasquirt-18/24496d1315591100-ms1-mspnp-closed-loop-ebc-works-well-my-car-details-pidcontrolbook2.pdf.

[80] Ashish Kumar Shakya, Gopinatha Pillai, and Sohom Chakrabarty. Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, 231:120495, 2023. ISSN 0957-4174. doi:

https://doi.org/10.1016/j.eswa.2023.120495. URL https://www.sciencedirect.com/science/article/pii/S0957417423009971.

[81] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[82] Taha Al-Saadi, J. Anthony Rossiter, and George Panoutsos. Control of selective laser melting processes: existing efforts, challenges, and future opportunities. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pages 89–94, 2021. doi: 10.1109/MED51440.2021.9480258.

[83] Ronan McCann, Muhannad A. Obeidi, Cian Hughes, Eanna McCarthy, Darragh S. Egan, Rajani K. Vijayaraghavan, Ajey M. Joshi, Victor Acinas Garzon, Denis P. Dowling, Patrick J. McNally, and Dermot Brabazon. In-situ sensing, process monitoring and machine control in laser powder bed fusion: A review. *Additive Manufacturing*, 45:102058, 2021. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2021.102058. URL https://www.sciencedirect.com/science/article/pii/S2214860421002232.

[84] Taha Al-Saadi, J. Anthony Rossiter, and George Panoutsos. In-situ proces control strategies for selective laser melting. *IFAC-PapersOnLine*, 56(2):6594–6599, 2023. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2023.10.357. URL https://www.sciencedirect.com/science/article/pii/S2405896323007243. 22nd IFAC World Congress.

[85] Stylianos Vagenas, Taha Al-Saadi, and George Panoutsos. Multi-layer process control in selective laser melting: A reinforcement learning approach. *Intelligent Manufacturing*, 2024. URL https://doi.org/10.1007/s10845-024-02548-3.

[86] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[87] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

[88] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Mohammad Ghavamzadeh, and Edgar A. Duéñez-Guzmán. Lyapunov-based safe policy optimization for continuous control. *ArXiv*, abs/1901.10031, 2019.

[89] Yongshuai Liu, Avishai Halev, and Xin Liu. Policy learning with constraints in model-free reinforcement learning: A survey. In *International Joint Conference on Artificial Intelligence*, 2021. URL https://api.semanticscholar.org/CorpusID:237100857.

[90] Quanquan Han, Heng Gu, Shwe Soe, Rossi Setchi, Franck Lacan, and Jacob Hill. Manufacturability of alsi10mg overhang structures fabricated by laser powder bed fusion. *Materials & Design*, 160:1080–1095, 2018. ISSN 0264-1275. doi: https://doi.org/10.1016/j.matdes.2018.10.043. URL https://www.sciencedirect.com/science/article/pii/S0264127518307986.

[91] Gabriele Piscopo, Alessandro Salmi, and Eleonora Atzeni. On the quality of unsupported overhangs produced by laser powder bed fusion. *International Journal of Manufacturing Research*, 15:1, 01 2020. doi: 10.1504/IJMR.2020.10019045.

[92] Hong-You Lin, Hong-Chuong Tran, Yu-Lung Lo, Trong-Nhan Le, Kuo-Chi Chiu, and Yuan-Yao Hsu. Optimization of surface roughness and density of overhang structures fabricated by laser powder bed fusion. *3D Printing and Additive Manufacturing*, 10(4):732–748, 2023. doi: 10.1089/3dp.2021.0180.

[93] Stylianos Vagenas and George Panoutsos. Constrained reinforcement learning for advanced control in powder bed fusion. In *2025 European Control Conference (ECC)*, pages 1828–1835, 2025. doi: https://doi.org/10.23919/ECC65951.2025.11186875.

[94] Zahra Fatemi, Minh Huynh, Elena Zheleva, Zamir Syed, and Xiaojun Di. Mitigating cold-start forecasting using cold causal demand forecasting model, 2023. URL https://arxiv.org/abs/2306.09261.

[95] Xuesong Gao, Fernando Okigami, Nicholas Avedissian, and Wei Zhang. An experimental and modeling study on warping in additively manufactured overhang structures. *Additive Manufacturing*, 81:104017, 2024. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2024.104017. URL https://www.sciencedirect.com/science/article/pii/S2214860424000630.