Article:
Mattera, G., Manoli, E., Canzini, E. et al. (2026) Integration of reinforcement learning in robotic additive manufacturing control: advances, challenges, and future perspectives. Journal of Manufacturing Processes, 169. pp. 202-241. ISSN: 1526-6125

https://doi.org/10.1016/j.jmapro.2026.04.069

Journal of Manufacturing Processes 169 (2026) 202–241

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

journal homepage: www.elsevier.com/locate/manpro

Integration of reinforcement learning in robotic additive manufacturing control: Advances, challenges, and future perspectives

Giulio Mattera [a,*], Elena Manoli [a], Ethan Canzini [b], Luigi Nele [a]

[a] Department of Chemical, Materials and Production Engineering, University of Naples "Federico II", Naples, Italy
[b] Space Instrumentation Lab, School of Electrical & Electronic Engineering, The University of Sheffield, Sheffield, UK

ARTICLE INFO

Keywords:
Generative AI
Reinforcement learning
Additive manufacturing
AI-integrated robotics systems
Process control

ABSTRACT

Reinforcement learning (RL) is emerging as a promising route to adaptive, data-driven control in robotic additive manufacturing (AM). This review surveys the literature on both traditional and RL-based controllers in AM and finds that, in the case of RL applications, most studies prioritise toolpath optimisation or process-parameter tuning over robotic feedback control, such as motion-platform actuation or power-source regulation, and that the majority remain confined to simulation studies. Only a small, though growing, subset attempts closed-loop control of melt-pool dynamics, bead geometry, or thermal profiles—objectives that are central to the production of high-quality AM parts. Where evaluated, RL demonstrates the ability to exploit high-dimensional sensing to manage nonlinear, multivariable interactions, thereby improving tracking performance, robustness, and part quality.

Following an in-depth examination of the state of the art in both traditional controllers and RL applications, this survey identifies and analyses the main barriers to industrial deployment, including the lack of formal stability and safety guarantees, sim-to-real mismatch and sample inefficiency, closed vendor platforms with limited low-latency actuation, and millisecond-scale real-time constraints on edge hardware. Finally, potential solutions to these challenges are discussed, including hybrid RL-traditional control architectures, data-efficient learning with digital twins and reduced-order models, offline RL and human-informed initialisation to minimise on-machine exploration, explainable RL for policy transparency, and hierarchical integration that incorporates additional Artificial Intelligence (AI)-based software modules such as in situ monitoring and anomaly detection to move beyond set-point regulation. Exploring these solutions could harness the most advanced AI techniques in manufacturing to improve learning efficiency, enhance the quality and reliability of feedback controllers, and increase policy interpretability for certification purposes in AM.

1. Introduction

In recent years, the manufacturing industry has been increasingly driven to adopt advanced solutions capable of addressing complex challenges [1], with the primary objectives of improving efficiency, reducing material waste, and achieving high-quality standards of the final produced parts [1,2]. Although traditional techniques such as linear modelling, response surface methodology, statistical process monitoring using control charts, and classical control approaches like Proportional-Integral-Derivative (PID) controllers have long been employed in industrial manufacturing systems [4,5], recent advances in information and communication technologies (ICT) and computer science are gradually displacing these methods in favour of artificial intelligence (AI)-based alternatives. This transition is largely driven by the increasing availability of process data, as modern manufacturing environments are now highly instrumented and capable of generating vast volumes of information in real time [6]. In such data-rich ecosystems, traditional analytical methods often fall short in effectively capturing the underlying complexity and non-linearities inherent in manufacturing processes, as well as suffer from the high dimensionality of the data [7]. These methods typically rely on simplified linear or moderately complex input-output relationships, which may not be sufficient to model the intricate dynamics and high-dimensional feature spaces characteristic of real industrial systems. In contrast to traditional approaches, AI techniques offer more flexible and scalable solutions that can extract meaningful patterns and insights from large, heterogeneous

* Corresponding author.
E-mail address: giulio.mattera@unina.it (G. Mattera).

https://doi.org/10.1016/j.jmapro.2026.04.069
Received 14 September 2025; Received in revised form 6 February 2026; Accepted 22 April 2026
Available online 30 April 2026

datasets, enabling more accurate prediction, control, and optimisation.

Among the various fabrication technologies, additive manufacturing (AM) has emerged as an innovative and disruptive approach. Unlike conventional subtractive methods that remove material from a solid block, AM builds components in a layer-by-layer fashion, significantly reducing material waste (often referred to buy-to-fly ratio) and enabling the fabrication of geometrically complex parts [8,9]. Although initially referred to as a rapid prototyping technique, the considerable progress made in AM technologies has shifted its role towards that of a mature and widely adopted manufacturing process, which has several applications in biomedical, construction, automotive, defence and aerospace sectors. In particular, polymer-based AM methods such as fused deposition modelling (FDM) are now widely adopted even at the consumer level, allowing individuals to fabricate low-cost, customised components for repair, design, fashion, and personal use. Despite these advantages, AM processes, particularly those involving metals or large-scale robotic systems, are inherently complex and prone to producing defective or non-compliant parts due to numerous interacting variables. The physical complexity of those processes causes significant challenges in ensuring consistent quality and process stability across different applications and operating conditions.

For this reason, the AM sector is increasingly focused on enhancing operational efficiency. Leveraging recent advances in AI, particularly in areas such as robot learning, perception, control, and interaction, AM systems may now achieve levels of adaptability and autonomy that were previously unreachable. However, the application of AI in AM is not a new concept. In fact, several review articles can be found in the literature which cover recent applications of AI in different AM processes.

For example, Sampedro et al. [10] reviewed 86 studies in which researchers working in the field of FDM employed acoustic emission (AE), accelerometer and vibration sensors alongside both supervised and unsupervised learning techniques for fault detection and broader process monitoring. Rajendran et al. [11] examined 152 papers and highlighted other common applications of AI, such as optimisation of process parameters, generative design, and the use of optical monitoring to fine-tune those parameters. In the domain of metal additive manufacturing, Liu et al. [12] demonstrated how machine-learning models, including neural networks, have been applied to process modelling tasks in Laser Bed Fusion (LBF), for instance, correlating process parameters with density or mechanical properties such as fatigue resistance, with the aim of generating process maps for parameter optimisation. Similarly, Zhang et al. [13] showed how AE sensors, thermographic imaging and light-based monitoring techniques can detect defects in situ, while Morandi et al. [14] illustrated how analogous approaches may be transferred to laser additive manufacturing systems, also those based on Direct Energy Deposition (DED). Finally, the reviews by Mattera et al. [15] and Xia et al. [16] surveyed the use of welding-current and welding-voltage sensors (alongside the others mentioned before, such as optical and audible sound sensors) for process monitoring in arc-based additive manufacturing and discussed emerging trends in closed-loop control. In particular, they describe how machine learning (ML) can model the relationship between process parameters and layer geometry, whose shape evolves during deposition in DED-Arc processes due to the arc self-regulation principle, thereby enabling adaptive path planning and dynamic adjustment of the build process.

Although the use of ML has been widely reported for addressing various challenges in robotic AM, particularly in data-driven modelling for predictive analysis within Cyber-Physical Systems (CPS) [17] and Digital Twin frameworks [18,19], its application extends beyond that. As briefly mentioned above, ML techniques have also been employed for process diagnosis, enabling the detection of anomalies and classification of defects [20,21], as well as for prognosis, through the development of forecasting models to estimate future system states [22,23]. These capabilities have significantly contributed to the advancement of maintenance strategies, supporting both corrective and predictive condition-based maintenance [24,25]. However, one of the most critical aspects remains under active research: the decision-making layer, also referred to as prescriptive analysis. This phase involves determining the optimal corrective or adaptive actions to be taken based on the outcomes of diagnostic and prognostic assessments, thus closing the loop towards autonomous and intelligent manufacturing systems.

In this context, expert systems are still commonly preferred in industrial applications, whereby specific corrective actions are triggered in response to defined events. Alternatively, when setpoint-based control systems are developed, conventional control strategies, such as PID controllers, remain widely used. However, a promising branch of ML, known as Reinforcement Learning (RL) [26], offers new possibilities. Therefore, given the importance of control systems within the decision-making layer, Fig. 1 presents an innovative control scheme for AM. In this scheme, actions are selected not only based on sensor measurements, translated into scalar or vector quantities relative to a reference to compute the control error, but also through the incorporation of AI-based outputs when generating the control policy. This approach enables effective action selection in multi-input, multi-output nonlinear complex systems, by using ML not only for measurements from images or estimation of states not directly measurable, but also using the output of monitoring systems, such as the information about the defect that occurred, estimated by a defect detection module.

While RL has been extensively reviewed in manufacturing [26,27], its application to AM has not yet been comprehensively addressed. Considering the unique challenges of AM, such as its complex physical dynamics and high-dimensional process parameters, and the strong potential of RL to autonomously determine optimal actions based on data, this review paper aims to systematically examine the state of the art in this field. The analysis is based on publications retrieved from the Scopus database, considering both articles and proceedings papers. The structure of the paper is as follows:

- In Section 2, we provide a concise overview of control system development, additive manufacturing technologies and a more detailed explanation of RL, including fundamental concepts and recent developments.
- In Section 3, we conduct a structured literature review, identifying application areas, reported case studies, and RL techniques implemented in relation to robotic systems for AM.
- Finally, Section 4 presents a critical discussion on the current challenges, research gaps, and future perspectives for the integration of RL in robotic additive manufacturing environments.

2. Background

2.1. Control systems development

Any physical system that evolves over time can be modelled as a dynamic system. Its state $x_t \in X$, which may include variables such as layer geometry, melt-pool size characteristics, or temperature values, depends on its previous state $x_{t-1} \in X$ and on the control input $u_t \in U$, following a relationship described by the function f. This relationship is given by:

$$x_t = f(x_{t-1}, u_t). \tag{1}$$

In this assumption, the system can be defined as Markovian, and this function can be approximated with any continuous function by using both linear and nonlinear tools. Real systems are also subject to disturbances w_t, which capture unmodeled dynamics or external perturbations; therefore, a more complete model is described by:

$$x_t = f(x_{t-1}, u_t) + w_t. \tag{2}$$

Because the true state is often not directly measurable, state observers are often used, which are dynamical system that ingests sensor readings y_t and produces an estimate based on the previous estimate, control action and actual sensor reading:

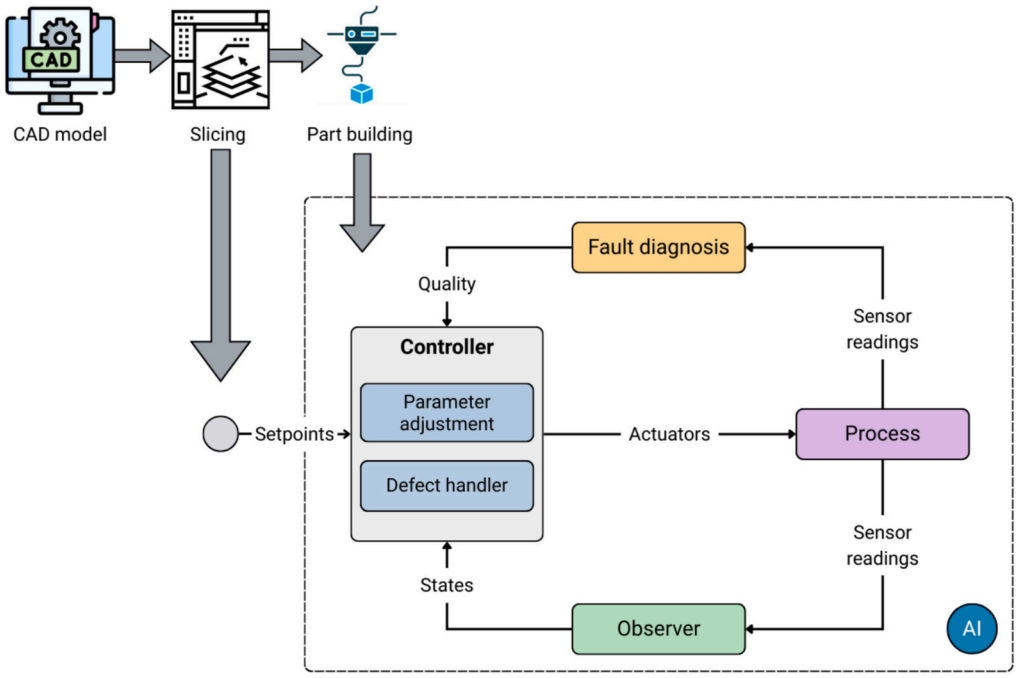

Fig. 1. Feedback control loop scheme for robotic additive manufacturing in which parameters are adjusted online or between layers based on the sensor's feedback, as well as by an AI-based monitoring system for anomaly and defect detection.

$$\hat{x}_t = g(\hat{x}_{t-1}, u_t, y_t) + w_t. \tag{3}$$

Therefore, the performance of a controller is largely affected by the goodness of the measurement. A closed-loop controller (see Fig. 2) is the software module that chooses the optimal control input u_t^* to minimise a cumulative cost function J, which encodes the desired performance (e.g. tracking error, control effort, safety margins), mathematically formalised in

$$\begin{aligned} u_t^* &= \arg\min_{u_t \in U} J \\ \text{subject to } x_t &= f(x_{t-1}, u_{t-1}). \end{aligned} \tag{4}$$

In traditional controllers, the J is usually a quadratic function with respect to the error and the effort of the controller, where the effort is quadratic with the amplitude of the action, where $\bar{x}$ is the tracking error ($\bar{x} = (\hat{x} - x_{ref})$). However, it is generally true that also simpler error-based controller can be used for a simple setpoint control task that incorporates quadratic cost on the states and inputs [29]:

$$J = \frac{1}{2}\left(\bar{x}R\bar{x}^T + uQu^T\right). \tag{5}$$

2.1.1. System identification

As discussed, accurate dynamic modelling of a process is essential for control-system design, since the control algorithm relies directly on the plant model. In many industrial applications, manufacturing processes are well approximated as Single-Input Single-Output (SISO) systems. For example, the melt-pool width in an additive manufacturing system may be assumed to depend linearly on laser power. Although a steady-state correlation between input and output can be readily identified, effective feedback control requires knowledge of the system's dynamics, whose dominant behaviour is often captured by first- or second-order linear models. By applying a step input of known amplitude (see Fig. 3), it is possible to characterise both the direction and speed of the process response.

In a first-order model (see Eq. (6)), the static gain K scales the input u to the output x, and the time constant τ defines the time needed to reach 80 % of the steady-state value.

$$G(s) = \frac{K}{\tau s + 1}. \tag{6}$$

On the other hand, a second-order model introduces two additional parameters: the natural frequency ω_n, which is the oscillation rate in the absence of damping, and the damping ratio ζ, which governs the rate at which oscillations dampen. With this dynamic model, in Eq. (7), it is also possible to model overshoot and oscillation around steady-state values:

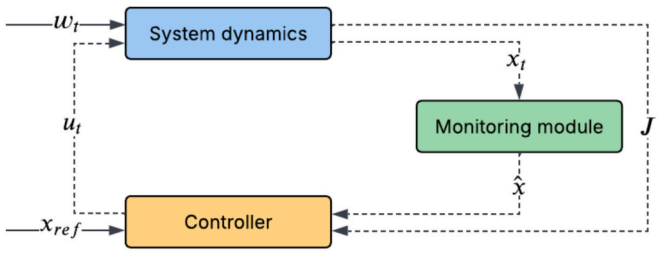

Fig. 2. A scheme of the main components of a control system.

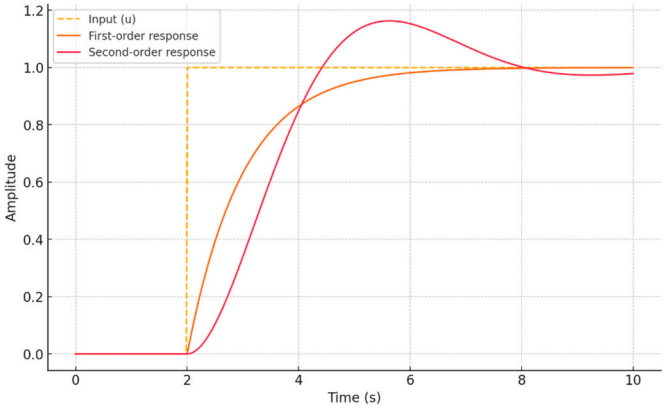

Fig. 3. Example of step response for first-order and second-order dynamic systems.

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n}. \tag{7}$$

The usage of these two additional parameters allows for the measurement of the peak-overshoot and settling-time metrics (e.g. the time to remain within ±2 % of the steady state). For a first-order system, settling typically occurs in approximately 3τ; for an underdamped second-order system, the 2 % settling time is roughly $4/\zeta\omega_n$.

Although the system identification rely directly on the step response of systems, some studies demonstrate how it is possible estimate steady state values of the system based on control action with the usage of ML models f_θ (see Eq. (8)), and then apply first order response, enabling the development of data-driven Reduced Order Model (ROM) of the systems [30]:

$$\widetilde{x}_{ss} = f_\theta(u). \tag{8}$$

In control-system design, the time-domain state-space form complements the transfer-function representation. A generic transfer function $G(s) = X(s)/U(s)$ can be expressed in the time domain as in Eq. (9) (approximate the time-derivative in a simple Euler-forward fashion), where x is the state vector, Δt the sampling period and u the scalar input. The A matrix (the system or state matrix) governs how the current states influence their own rates of change, encapsulating the process's internal dynamics (for example, natural damping or oscillation). The B matrix (the input matrix) specifies how the control input, such as laser power or wire-feed rate in metal AM, enters and drives each state equation.

$$x_{t+1} = Ax_t + Bu_t = x_t + \Delta t(Ax_t + Bu_t) \tag{9}$$

Generally, in first-order system $A = -\frac{1}{\tau}$ and $B = \frac{K}{\tau}$, while in second order system $A = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix}$ and $B = \begin{bmatrix} 0 & K\omega_n^2 \end{bmatrix}$.

2.1.2. Traditional feedback controllers

The most commonly employed control law for manufacturing systems is the PID controller (see Eq. (10)), which is supported by well-established methods for parameter tuning. Despite its popularity, tuning a PID controller can be challenging in practice. To address this, various AI techniques such as Genetic Algorithms (GA) have been applied. Nevertheless, PID controllers have inherent limitations, particularly their lack of adaptability to system dynamics. Some systems exhibit time-dependent or state-dependent linear dynamics, while others behave nonlinearly. These behaviours are difficult to manage with fixed PID control laws and typically require gain scheduling or, more recently, the use of neural networks to generate PID constants that can better accommodate such complex dynamics [31].

$$u_t = K_p e_t + K_i \int e_t + K_D \frac{de_t}{dt} \tag{10}$$

where $e(t)$ is the error between the current state and the reference state. PID control is one of the simplest yet, with a few extensions, most powerful feedback laws. Its primary limitation, however, arises when the controller output saturates, either due to actuator limits or safety constraints. In practice, the actuator command must be clamped to prevent mechanical damage or unsafe operation. But when the output is saturated and the setpoint is not reached, the controller's integrator continues to accumulate error. This “integral windup” can lead to large overshoots and sluggish recovery once the actuator unsaturates. To prevent this, anti-windup schemes (for example, conditional integration or back-calculation) are routinely incorporated. Beyond single-loop SISO systems, PID can be augmented with gain scheduling to handle systems whose dynamics vary across operating points: the controller gains are adjusted online based on measurable parameters, effectively treating the plant as locally linear around each setpoint. The second major challenge occurs in multivariable (MIMO) systems. In a typical MIMO plant, each input–output channel may interact, represented by nonzero off-diagonal transfer functions G12 and G21 in Eq. (11). A naive application of independent PID loops fails to account for these cross-couplings. Instead, more sophisticated multivariable controllers (e.g., LQR, or MPC) are required to allocate control actions across all channels simultaneously.

$$\begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix} = \begin{bmatrix} G_1(s) & G_{12}(s) \\ G_{21}(s) & G_2(s) \end{bmatrix} \begin{bmatrix} u_{1t} \\ u_{2t} \end{bmatrix} \tag{11}$$

As discussed in the control-system design section, the controller's objective is to solve Eq. (4). Therefore, by treating x as the state vector (with N components) and u as the control-input vector (with M components), the weighting matrices Q and R can be chosen as N × N and M × M, respectively. If the plant is linear and can be represented by a first-order transfer function, the Riccati equation can be solved over an infinite time horizon, yielding the optimal feedback law $u_t = -Kx_t$, where K is the gain matrix computed from the solution of that Riccati equation [32]. While the presented controller, named Linear-Quadratic Regulator (LQR), is elegant and computationally efficient, it comes with several practical limitations, with the three main ones related to (i) supposing linear dynamics, (ii) no explicit constraint handling and (iii) full state feedback requirement. LQR assumes that any control action can be applied. In practice, however, real systems are subject to hard bounds, and violating these constraints can damage hardware or lead to unsafe operation. Moreover, the system dynamics should be linear, and it is mandatory to measure (or estimate) all state variables. Although the LQR yields a single, fixed feedback gain that is optimal only under idealised, unconstrained, infinite-horizon, linear assumptions, it proves inadequate for complex, nonlinear systems. To address this limitation, researchers have adopted a receding-horizon approach, giving rise to Model Predictive Control (MPC) [33]. MPC preserves the quadratic-cost philosophy of LQR but replaces the static solution with an online optimisation at each sampling instant, leveraging model-based forecasts to determine the best control action. Essentially, MPC operates over a finite prediction horizon: at each control step, the controller simulates the system's evolution for a prospective sequence of inputs and solves a constrained quadratic programme to minimise the cost. This generates a trajectory of open-loop optimal control inputs starting at the current time t for the duration of the prediction horizon T

$$\boldsymbol{u}^* = \{u_t^*, u_{t+1}^*, u_{t+2}^*, \ldots, u_T^*\}. \tag{12}$$

For the closed-loop implementation, only the first input of the optimised sequence is implemented; the horizon then “recedes,” and the entire process repeats using updated state measurements. This strategy allows explicit enforcement of input and state constraints and readily accommodates nonlinear models, including those learned via ML techniques [34], to predict and control future system dynamics. The key part of the MPC controller is the optimisation step that finds the optimal control signal u_t^* at each timestep by minimising the cost function

$$u_t^* = \arg\min_{u \in \mathscr{U}} \left\{ C(x,T) + \sum_{t=1}^{T-1} C(x,t) \right\}, \tag{13}$$

where $\mathscr{U}$ is the set of all possible controls, $C(x,t)$ is the cost at each associated state and timestep and $C(x,T)$ is the terminal cost when the system reaches the final state. These costs can be formulated depending on the system, but generally they are kept as quadratic costs to aid in the computation of the optimal control inputs [35]. Additionally, MPC allows the formulation of the optimisation step to include constraints on both the state and the control inputs [36], which can form the joint polyhedral set $(x,u) \in \mathscr{Z}$, where

$$\mathscr{Z} \triangleq \{(x,u) : \forall i \in \mathscr{I}_{\mathscr{Z}}, c_i^{\mathrm{T}} x + d_i^{\mathrm{T}} u \leq 1\}. \tag{14}$$

MPC has long been a part of the key controllers used in industry, thanks to its ability to handle disturbances and constraints and for being able to provide optimal inputs for systems across manufacturing [37],

agriculture [38] and robotics [39]. However, the main limit of MPC in comparison to the offline computation for LQR is the computational time needed for running the simulations of the dynamics to predict what the effect of specific control inputs will be on the performance of the system. Continuously running an optimisation problem on a model of the system until convergence is achieved can be dependent on the dimensionality of the state space and the number of optimisation steps that are taken, which in turn can lead to increased computation time for the system. Additionally, a key assumption for many MPC problems is that the system being controlled is linear time-invariant (LTI). For many real-world systems, the actual dynamics are often nonlinear and control-affine

$$x_{t+1} = f(x_t) + g(x_t)u, \tag{15}$$

and thus would require linearisation to ensure they are compatible with the problem formulation. Furthermore, a model in simulation is only an approximation of the real-world model and thus may not capture the effective dynamics or disturbances the system may possess.

To address these problems, the field of MPC research has leveraged advancements in both the field of probabilistic programming and machine learning to either approximate or speed up the solution convergence for MPC problems. A key point is the use of convex optimisation, in particularly the alternating direction method of multipliers (ADMM), to speed up the optimisation process within the MPC control loop [40]. The use of convex optimisation has also found its use in splitting methods, which combines ADMM with proximal optimisation to solve a convex quadratic optimisation problem, which can be solved efficiently and quickly, and a set of single period optimisation problems that are solved in parallel, thus allowing the MPC problem to be implemented efficiently with no division operations [41]. An alternative to these approaches is using machine learning to learn more efficient models of the dynamics of the system, thus allowing complex nonlinear dynamics with disturbances to be modelled using lower-dimensional representations. Gaussian Processes (GPs) allow for the capturing of nonlinear dynamics through various kernels and can output linear estimations, which can speed up the optimisation step within the control loop [42]. It has also shown that GPs can be embedded within the controller directly and can be used to evaluate the stability of the controller whilst maintaining data-efficiency, a key improvement when using data that has been collected from a real system [43]. An additional benefit of this approach is that the uncertainty of the system can be quantified as part of the disturbances of the system, allowing the analysis of the dataset that is used to build the GP model. This can be extended to noisy data, allowing the combination of state estimation and control of nonlinear systems whilst maintaining the linear requirements for the optimisation problem [44]. More recently, the topic of learning models from data has led to a new breed of MPC approaches that incorporate probabilistic models into the optimisation step of the MPC controller. This has shifted the conversation of MPC from a *model-based* controller to a *policy-based* controller, where the optimisation step is sped up through the use of a parameterised policy that can consist of a static distribution, a set of distributions or a neural network [45]. The propagation of these approaches in literature has led to the coupling of RL and MPC approaches by approximating solutions to the Bellman equation, thus allowing controllers to interface with their environment and learn a transition model that is able to be optimised efficiently [46].

2.1.3. Stability guarantees

In general, most MPC methods can guarantee optimality through convergence properties when performing the optimisation step. However, the question remains whether these are locally optimal or globally optimal solutions. To find globally optimal solutions across the entirety of the state space $\mathscr{X}$ and the control space $\mathscr{U}$, one must examine the cost-to-go or value function over the given trajectory

$$J^*(x,t) = \min_{u\in\mathscr{U}}\left\{ C(x,T) + \int_t^T C(x,\tau)d\tau \right\}. \tag{16}$$

This in turn gives rise to the Hamilton-Jacobi-Bellman (HJB) equation, which allows the computation of optimality for any control input across the entire trajectory [47].

$$-\frac{\partial J^*(x,t)}{\partial t} = \min_{u\in\mathscr{U}}\left\{ C(x,\tau) + \frac{\partial J^*(x,t)}{\partial x}[f(x) + g(x)u] \right\}. \tag{17}$$

This however poses a high-dimensional nonlinear problem, as finding the optimal value function J^* is not a trivial matter. A variety of methods exist for approximating this solution, including nonlinear optimisation, dynamic programming and using neural networks as approximators for solutions [48]. An alternative would be to employ other optimality criterion such as Pontryagin's Maximum Principle (PMP) and shorten the optimality to local states only [43].

Another key aspect of control theory is the notion of stability. When designing feedback controllers, stability implies that the feedback loop will not cause the plant to go towards unstable states which are unrecoverable. For traditional approaches such as PI controllers, this involves choosing the location of poles and zeros such that

$$\lim_{t\to\infty} e_t = 0. \tag{18}$$

For MPC or optimal control methods, the use of closed-loop feedback after finding the optimal control input allows for stable responses, albeit requiring tuning of the optimisation parameters. By definition, the infinite horizon case of MPC implies stability for linear systems and reduces down to the LQR controller [35]. This in turn can be extended to nonlinear systems and finite horizon cases by applying receding horizon optimisation and Lyapunov functions, thus allowing them to retain stable feedback laws despite the nonlinearity. For the probabilistic methods that use machine learning to approximate either the dynamics or the controller, a key result for many state space systems are their similarity to Markov chains. As they retain the memoryless property, theses probabilistic state spaces contain an equilibrium measure $\pi(\mathscr{X})$ that implies the existence of a steady-state in the chain, thus allowing the feedback controller to be designed such that [49].

$$\lim_{t\to\infty} \mathbb{P}^t(x,u) = \pi(\mathscr{X}), \tag{19}$$

thus allowing probabilistic models to be combined with controllers to drive systems towards desired reference points. The use of Markov chains, and in particular the formulation of Markov Decision Processes, allows the application of stochastic control and reinforcement learning to many optimal control problems, the latter of which will be discussed in more detail in Section 2.4.3.

2.2. Traditional control theory applied to additive manufacturing processes

AM technologies may be classified by their energy source into five main families: laser-based, extrusion-based, material-jetting, electron-beam and arc-based processes (see Fig. 4). In what follows, we first summarise the operating principles of principal technologies, identify the key process parameters and discuss how those parameters influence final part quality. Then, for each technology, it is described why closed-loop feedback control is essential and representative applications that employ conventional (non-RL) control strategies are discussed. This overview will pave the way for a subsequent survey of cutting-edge studies that integrate RL methods into AM process control.

2.2.1. Laser-based additive manufacturing

Laser-based AM processes employ a medium-to-low-power laser beam to effect phase changes in material, either by melting solid feedstock or by curing a photosensitive resin. In the case of laser melting, the

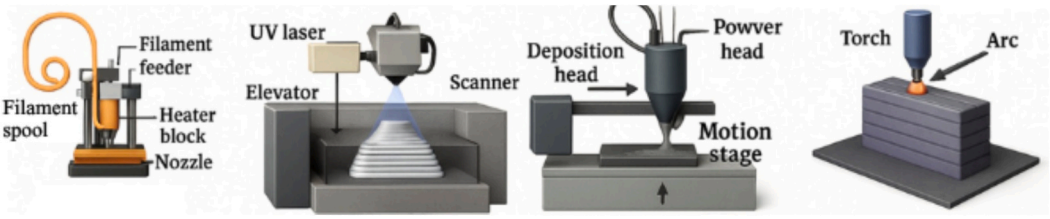

Fig. 4. Four typical AM technologies: (a) fused deposition modelling (FDM), (b) stereolithography (SLA) (c) laser metal deposition (LMD), and (d) wire arc direct energy deposition (WA-DED).

material may be supplied as metal powder to a pre-spread powder bed, where it is selectively melted and solidified to build up the part (commonly known as Selective Laser Melting (SLM), within the powder-bed fusion family). Alternatively, feedstock can be delivered directly into the melt zone, and in particular when powder is blown through a nozzle into the laser focus, this is referred to as powder-based Laser Metal Deposition (P-LMD), whereas when a metal wire is fed instead, it is called wire-based Laser Metal Deposition (W-LMD).

By contrast, laser polymerisation processes work with a liquid, photosensitive resin that undergoes rapid curing upon exposure to a low-power ultraviolet laser. Stereolithography (SLA) exemplifies this approach, using a focused UV laser to trace and solidify resin layer by layer, and it is a technique used for polymer printing.

Regarding the materials that can be printed by employing those technologies, when metal powders are used, such as in SLM and P-LMD processes, the most commonly used feedstocks are stainless steels (notably 316L), titanium alloys (especially Ti-6Al-4V), aluminium alloys (for example AlSi10Mg), nickel-base superalloys (such as Inconel 718 and Inconel 625), cobalt-chromium alloys (CoCrMo) and various tool steels (including maraging grades and H13) [50]. Conversely, when wire is employed, as in wire-based laser metal deposition, the use of readily available welding wires makes it possible to process a much broader range of alloys without the need for bespoke powder formulations [51]. Since the production of metal powder is substantially more expensive than that of welding wire, W-LMD provides a distinct cost advantage. Furthermore, wire-fed systems are not constrained by the build-chamber dimensions typical of powder-bed machines, which allows the production of considerably larger components. This also reduces the handling risks and safety concerns associated with fine metal powders.

In laser AM, porosity size, surface roughness, and overall process stability are controlled by key melt-pool characteristics, namely, width, depth, and thermal gradient, which dynamically respond to the controllable inputs such as laser power, powder mass flow or wire feed rate, and scan speed [52–54], so on-the-fly adjustments can stabilise the deposition and minimise defects. Research has demonstrated that porosity decreases with increasing laser power, and that layer geometry shifts accordingly. In fact, higher scan rate mainly shrinks melt-pool depth, while boosting laser power enlarges and deepens the pool and raises its temperature, with power having the strongest effect on depth. Moreover, the process parameters impact the mechanical properties as well [55,56].

Given the importance of the melting pool area in powder-based L-AM, several studies have been conducted to develop a feedback controller that is able to adjust the laser power to control the melting pool area. Haussain et al. [57] developed a physical model for the melting pool area and used a PID to control the laser power in a simulation environment. Liu et al. [58] used a camera, a second-order system transfer function, and a PID controller for a similar purpose, validating their methodology on a real system. Akbari et al. [59] instead focused on molten pool width, using a first-order model linked to layer geometry rather than thermal stability. Beyond pool characteristics, layer height is also a key factor in AM; therefore, Shi et al. [60,61] proposed a similar methodology (system identification and PID controller) to control the layer height in a P-LMD process by modifying the laser power based on external setpoints, enabling to produce with reduced amount of layers some geometries with height variation along deposition path. In laser metal deposition with wire feed (L-LMD), where melting introduces distinct thermal and fluid-dynamic phenomena, tailored controllers have been developed to address its specific control requirements. For instance, Bernauer et al. [62] employed a PI controller that, after each layer, adjusts the wire feed speed for the subsequent pass to restore the measured layer height to its nominal value. Despite this study, no previous work has considered the multi-input nature of the process. Liu et al. [63], however, developed an MPC controller based on open-loop step tests of laser power. They measured both melt-pool area and width via near-IR imaging, fitting two first-order-plus-delay transfer functions (one for area, and one for width) to capture the SIMO dynamics with a single input, the laser power and multiple outputs, both melt-pool area and width. It is evident from the state of the art that linear modelling and PID controllers are primarily used to regulate the melt-pool's width or area during powder-based, thereby enhancing process consistency. This procedure is illustrated in Fig. 5**.** Yet, variations in input power can introduce defects, indicating the need for more advanced feedback laws that not only enhance final part quality (e.g., by reducing porosity and surface roughness) but also provide tighter control of melt-pool dynamics. Traditional control schemes lack both the ability to process complex data streams, such as real-time imagery and the flexibility to address these multifaceted objectives, especially given that other parameters, such as the scanning speed, can also be varied.

Regarding SLA process, resins are typically based on epoxies, acrylates or thermoplastic elastomers [64]. In this process, the primary material-related parameters are the laser's exposure time and light intensity, while the key scan-path variables include hatch spacing (overlap) and layer thickness. Because photopolymerisation inherently induces volumetric shrinkage, printed parts often suffer from dimensional distortion and surface irregularities. By carefully tuning exposure conditions and scan strategy, it would be possible to mitigate shrinkage effects, thereby enhancing dimensional accuracy, reducing surface roughness and improving overall part quality. For this reason, as in the case of metal laser AM, in-situ and real-time process monitoring applications alongside real-time feedback control would be useful to improve the 3D printing quality, but to the authors 'knowledge, traditional controllers have not been applied for this scope in the field of SLA.

2.2.2. *Electron-beam-based additive manufacturing*

In the family of PBF techniques, electron-beam melting (EBM) replaces the laser with an electron beam, while its DED counterpart feeds wire rather than powder spread on the bed [65,66]. EBM has been successfully applied to aerospace alloys such as Ti-6Al-4V, Inconel 625 and Inconel 718, producing fully dense parts with mechanical properties comparable to those of wrought material. However, microstructural variations can arise from thermal gradients within the build chamber, highlighting the importance of feedback control. In both powder- and wire-fed systems, the energy input is determined by the accelerating voltage, beam current (or beam power), and scan speed, and in wire-DED systems additionally by the wire-feed rate. While continuous EBM delivers a steady heat input, pulsed EBM adjusts the beam's duty cycle to control average power, offering finer thermal management,

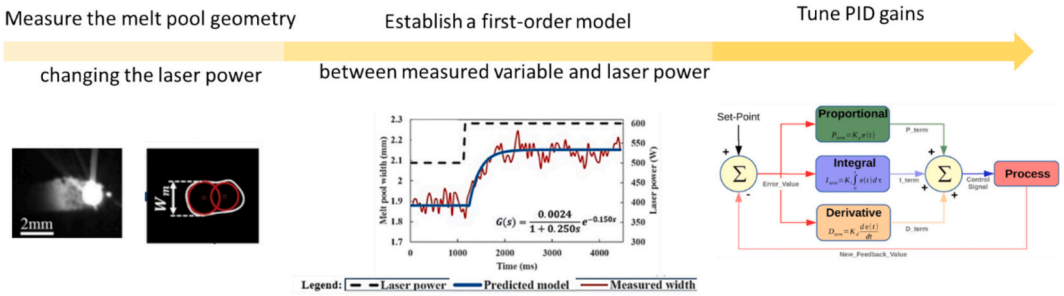

Fig. 5. The state of the art of traditional feedback control in laser-based AM shows that it consists of measuring the melt pool geometry (area or width) at changing the laser power with a camera and then tune a PID controller once a first or second order transfer function is fit on dynamic response.

enhanced melt-pool stability and reduced residual stress [67]. As in the case of LMD, feedback control laws have been developed to dynamically adjust the process parameters in real time or between layers, depending on the closed-loop control goal.

Vasileska et al. [68] developed a regression-based expert system that correlates the beam-duty cycle with the melted area combined with a feedback controller using image processing. and an expert system adjusts the duty cycle to maintain a consistent melt area between layers, improving the overall stability of the printing process. An example applied to Wire-EBM is the work of Liang et al. [69], in which the authors identified a first-order dynamic model relating melt-pool width to beam current and used PID control to regulate pool width online, allowing for dynamic adjustment for varying layer geometry during the deposition. In both studies, similarly to the works previously discussed in the field of laser-based AM, real-time camera monitoring of the melt pool was essential to their closed-loop control strategies, enabling the application of traditional controllers based on system identification and SISO control. However, as Liang et al. also demonstrated, additional factors such as beam travel speed significantly influence layer geometry, and an inappropriate combination of process parameters can lead to defects such as porosity or incorrect heating, defects that traditional control strategies often overlook. This underscores the need for more sophisticated, multi-variable feedback controllers in EBM-based AM, as well as in laser-based AM. Simply adjusting the duty cycle or beam current allows control over melt-pool area and width, but the dynamic interactions among other system states, such as scan speed, wire-feed rate and thermal history, remain unmonitored and uncontrolled. To achieve truly stable, defect-free builds, advanced control architectures must concurrently track and regulate multiple aspects of the deposition process rather than treating each parameter in isolation.

2.2.3. *Extrusion-based additive manufacturing*

Material extrusion employs a heated nozzle to soften or melt plastic filament, which is then extruded as a continuous bead that rapidly cools and solidifies to form the intended geometry. In fused deposition modelling (FDM), a dual-nozzle print head deposits molten thermoplastic, heated roughly 1 K above its melting point, onto the build platform, where it fuses to preceding layers. The affordability, non-toxicity and odourless nature of common filaments (e.g. ABS, PLA, medical-grade ABS, casting wax and elastomers) combined with desktop-scale hardware make FDM ideal for rapid prototyping and low-cost parts, although its dimensional accuracy and surface finish remain inferior to those of powder-based polymer AM. FDM can process a wide array of materials, from standard polymers (ABS, PLA, Nylon, ASA, PETG, PC) to high-performance plastics (PEEK, PEKK, ULTEM), flexible thermoplastic elastomers and composites reinforced with carbon nanotubes, graphene, metal or ceramic fillers, or continuous and short fibres [70]. Parts often suffer from residual-stress-induced distortion, microvoids between beads, uneven fibre distribution, weak adhesion to the plate and high surface roughness. These defects arise from local variations in thermal history and bead geometry but can be mitigated by online adjustment of nozzle temperature, extrusion rate, print speed, layer thickness and raster angle. Closed-loop controllers, using thermal, optical or acoustic sensors which detect process drifts in real time, can be used to compensate for those errors [71]. In FDM, the most critical parameters that must be regulated via real-time feedback control are the extruder temperature, the flow rate and the build-plate temperature. Maintaining precise thermal conditions is essential to ensure a consistent polymer melt flow rate, even in the presence of disturbances such as partially not melted polymer chunks or transient blockages. If temperatures deviate, the melt viscosity can change unpredictably, leading to flow-rate fluctuations that produce defects in the deposited filament. Moreover, incorrect thermal settings can cause thermal degradation of the polymer, compromise the mechanical performance and structural integrity of the final part, introduce dimensional inaccuracies, and generate inhomogeneities in the extrudate. Robust, closed-loop temperature control is therefore indispensable for achieving high-quality, repeatable FDM builds. Despite the underlying complexity of FDM, researchers have demonstrated that the flow dynamics can be linearised sufficiently to allow the application of classical PID controllers, augmented with anti-windup schemes, for regulating a fixed set-point flow rate [72]. However, this approach can be further updated to better respond to nonlinearities and unmodeled disturbances inherent in additive manufacturing through AI methods, such as neural networks used in PID gains tuning [73]. Most recently, convolutional neural networks (CNNs) have been trained to analyse live camera images of the extrudate, estimate its instantaneous flow rate, and then adjust the reference flow rate in a simple linear fashion at this stage, the limitations of pure policy gradient methods such as REINFORCE become evident, particularly due to the high variance associated with Monte Carlo return estimates. A natural extension is the Actor-Critic framework, in which two neural networks are trained simultaneously with distinct roles [74]. Collectively, these advances indicate a new generation of FDM controllers that transcend traditional sensor-PID-actuation loops. By using complex feedback, such as image, control actions can be computed directly from the images, something conventional controllers are not equipped to do. However, this possibility has not been explored yet for FDM.

2.2.4. *Arc-based additive manufacturing*

Arc-based AM is rapidly gaining traction for large-scale metal part fabrication because it leverages conventional arc-welding equipment to build up material layer by layer at a fraction of the cost of powder-based systems [75,76]. In wire arc-DED (WA-DED), both gas metal arc welding (GMAW) and plasma arc welding (PAW) can be used, and a wide variety of readily available welding wires make the process highly flexible and economical [77]. The main drawback of WA-DED, though, is its relatively poor as-built surface finish compared with other AM methods, typically necessitating substantial post-processing [78]. Like all AM processes, WA-DED relies on precise heat input and stable melt-pool behaviour to achieve consistent layer geometry [79,80]. In many powder-based techniques, changing a process parameter predictably

alters the melt-pool size and thus the deposited bead dimensions. Although this is also valid for WA-DED [81,82], it presents additional challenges. Even with fixed welding parameters, the cross-sectional shape of each bead can drift over a build as the geometry evolves [83]. In addition, arc current and voltage fluctuate stochastically [84], causing unpredictable heat input. Although advanced waveform-controlled methods, such as cold-metal transfer (CMT) [85], greatly improve arc stability, they cannot fully compensate for geometry-induced variations. Without active adjustment, these combined effects can lead to layer collapse in thin features, over- or under-fill in transitional sections, and increased porosity or cracks. Consequently, closed-loop feedback control of key parameters such as welding voltage, wire feed speed, torch distance from substrate and welding speed is essential. To that end, recent studies have proposed a variety of in-process sensing and control architectures, ranging from vision-based melt-pool monitoring to adaptive current profiling, to maintain consistent deposition quality throughout a WA-DED build.

Given the critical importance of closed-loop control in WA-DED, a growing body of research has aimed to model and regulate its key process variables. Xiong et al. [86] developed a first-order dynamic model relating wire-feed speed (WFS) to contact-tip-to-workpiece distance (CTWD) and showed that adjusting WFS in real time can compensate for CTWD drift, reducing melt-pool instabilities and defects. Building on this, Xia et al. [87] used camera-based measurements of bead width to drive an autoregressive-exogenous (ARX) model with MPC, successfully correcting width deviations that otherwise lead to a collapse in thin walls. However, both of these approaches treat WA-DED as single-input single-output (SISO) systems, controlling only arc length or bead width, so they cannot simultaneously regulate bead height and width. To address multi-output needs, Wang et al. [88] proposed two active-disturbance-rejection controllers (ADRC): one for height and one for width. They fused their outputs by weighting the welding-current commands (90% from the width controller, 10% from the height controller, then the current is controlled by adjusting WFS using a PID) but still ignored the intrinsic coupling between these two geometrical states. More recently, Mu et al. [89] demonstrated a full MIMO MPC scheme that uses simultaneous camera measurements of both bead width and height to optimise deposition quality. Beyond vision-based sensing, emerging studies have explored inferring bead geometry and CTWD variations directly from arc-current and voltage signals [90] or audio [91], promising faster, in-process feedback without line-of-sight constraints. Nevertheless, true MIMO control of WA-DED remains underexplored: stochastic arc fluctuations, nonlinear thermal–mechanical interactions, and competing objectives (geometry, energy efficiency, defect suppression) exceed the capabilities of traditional linear controllers.

2.3. Limits of traditional controllers applied to additive manufacturing

As demonstrated across all the AM systems presented, the state-of-the-art in feedback control involves adjusting process parameters on-line or between successive layer depositions using traditional controllers, most commonly PID, derived via linear system identification. The principal objectives (see Fig. 6), especially in metal additive manufacturing, are to regulate melt-pool dimensions in powder-based processes to enhance stability, and to correct the CTWD in wire-based processes to maintain a consistent heat-input profile during deposition and thereby avert defect formation. Other applications focus on controlling layer geometry: by modulating processing parameters, one can compensate for disturbances arising from the system's complex dynamics, enabling on-the-fly adjustment of deposition rates, accelerating build times and improving the quality of the final components.

To date, most studies address only one response variable at a time, simplifying the feedback loop to remain compatible with conventional controllers. In practice, AM processes are inherently nonlinear and multivariable. In addition to power-source settings such as energy input, parameters like CTWD, deposition speed, and layer overlap can also be varied, often through coordinated robot-arm movements. These variables, however, are rarely controlled simultaneously. The reliance on single-input approaches leaves significant interactions unmanaged and can reduce overall performance; in most implementations, all but one parameter is fixed while a single input is varied, implicitly assuming the others play no active role. This neglects the fact that each parameter influences energy consumption profiles [92] and thermal behaviour differently [93], and that manipulating one parameter in isolation can precipitate unforeseen deposition anomalies. While such single-parameter control schemes represent a clear advance over open-loop operation, appreciable gains remain attainable. Accordingly, in the next section we introduce the principles of RL, explaining why it offers a compelling route for AM control from the authors' perspective and surveying current state-of-the-art techniques. Thereafter, we undertake a detailed review of recent AM research, analysing all pertinent studies and comparing their merits to those reported above.

2.4. Reinforcement learning

Reinforcement learning (RL) [94] provides a model-free learning framework for sequential decision making under uncertainty, ideally suited to the complex, nonlinear and time-varying dynamics, such as those encountered in AM systems. As depicted in Fig. 7a, an RL agent learns a control policy via trial-and-error interactions with the process, either directly on the physical or within a reduced-order model (ROM) of the system. The policy may first be trained in simulation using the ROM

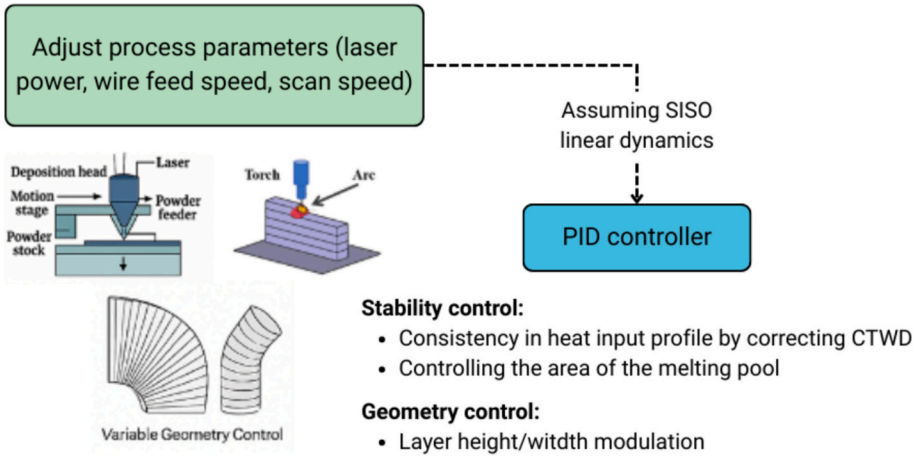

Fig. 6. Schematic overview of feedback control in additive manufacturing.

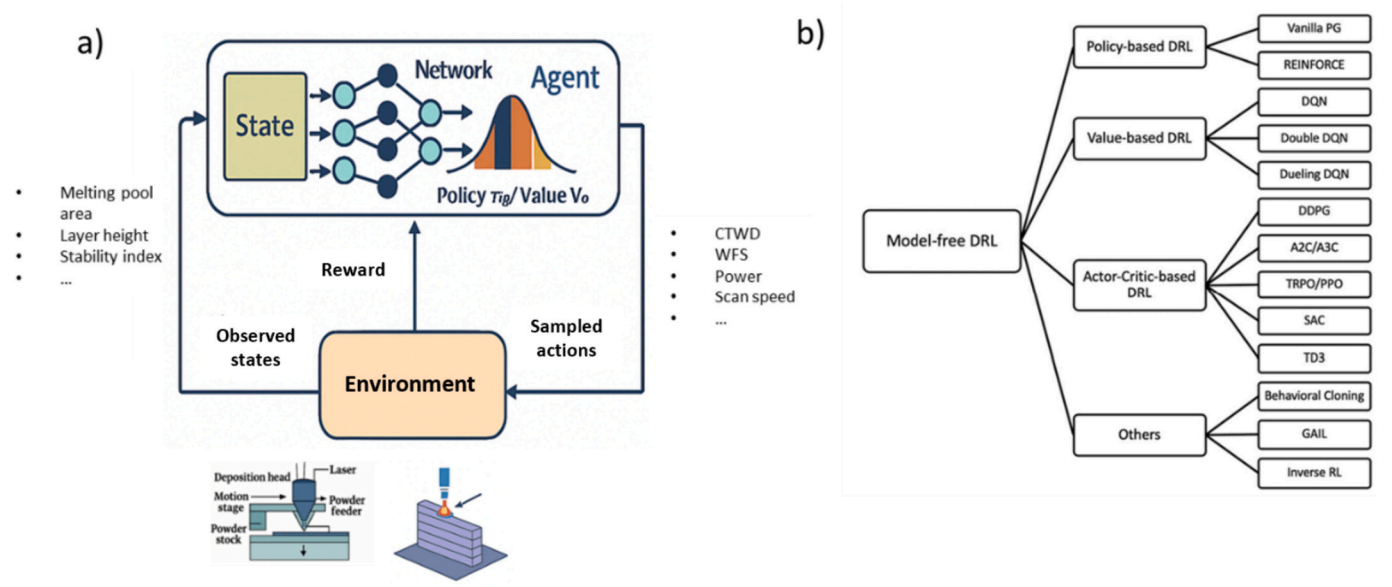

Fig. 7. **(a)** Workflow of a deep-reinforcement-learning agent interacting with an AM process: system states (computed by raw sensors or estimated by ML models) are fed into a neural-network policy/value estimator, which samples actions (process-parameter adjustments) and receives rewards from the environment. **(b)** Classification of model-free DRL methods.

to accelerate convergence, then subsequently fine-tuned on real-world data to ensure robust performance in the actual AM environment. RL can address limitations of traditional PID control, and it adaptively optimise melt-pool stability, heat input, and deposition geometry by simultaneously using all the system states and all the possible actions (adjustable parameters).

RL can be broadly classified into Model-Based RL and Model-Free RL, depending on whether a model of the system dynamics, typically expressed in terms of state transition probabilities, is known and, more importantly, whether such a model is explicitly used to determine the optimal control action.

In Model-Based RL, the agent relies on a representation of the system dynamics, either assumed to be known a priori or learned from data and exploits this model to plan future actions. While this approach is theoretically appealing due to its potential data efficiency, its practical application in manufacturing environments is limited. Industrial processes, including AM systems, are characterised by strong nonlinearities, time-varying behaviour, process disturbances, and uncertainties that are difficult to capture accurately in a predictive model. As a result, the agent is required to simultaneously learn both the system model and the control policy, leading to increased computational complexity and a higher risk of performance degradation due to modelling errors. For these reasons, Model-Free RL is far more commonly adopted in practical manufacturing control applications. In this setting, the agent does not attempt to explicitly model the system dynamics but instead learns a control policy directly through interaction with the environment. Once the control problem is properly formulated, in terms of state representation, action space, constraints, and reward function, drawing on established principles from control theory and process knowledge, the learning task can focus exclusively on policy optimisation. This separation significantly simplifies the learning problem and avoids the need to infer an accurate dynamical model of the plant.

From an industrial perspective, it is generally more effective to invest effort upfront in the design of the environment and control formulation, rather than attempting to learn both the system dynamics and the control policy online through direct interaction with a real manufacturing process. Consequently, despite the conceptual advantages of Model-Based RL, Model-Free RL represents the dominant and most practical paradigm for RL-based control in manufacturing applications. Accordingly, the remainder of this section focuses on Model-Free RL approaches.

At its core, RL begins by randomly initialising a policy that maps the current process state to an optimal control action. At each time-step, the agent selects an action based on its policy; the AM environment then evolves to a new state in response, yielding a scalar reward computed by a predefined function of the state and/or state-action pair. This reward signal is fed back to update the policy, strengthening decisions that lead to higher returns. The simplest RL algorithm, Q-learning, stores a tabular Q-value (which depends by the reward) for each state-action combination and greedily selects the action with the highest value. However, in AM the state and action spaces are often continuous and high-dimensional (e.g. laser power, scan speed, CTWD, and melt pool geometry). Modern approaches therefore employ deep neural networks to approximate the policy and, in some cases, the value function. Deep RL thereby scales to multivariate, continuous controls, though in some cases expert knowledge or hybrid schemes (e.g. discretised Q-learning) can still prove effective. Before discussing specific AM applications, where the definitions of states, actions, and reward functions must be carefully tailored, we first outline practical considerations for implementing such agents in software.

2.4.1. *Elements of reinforcement learning*

According to Sutton and Barto [94], an RL problem is defined by the interaction between an agent and its environment, characterised by four fundamental elements: a policy, a reward signal, a value function, and, in some cases, a model of the environment. This interaction is illustrated in Fig. 8, which depicts the standard agent-environment loop underlying RL algorithms. In control-oriented applications, the agent corresponds to a decision-making controller, while the environment represents the physical system or process being regulated.

As shown in Fig. 8, the agent receives measurements of the system state and selects actions according to its current policy. Let $x_t \in \mathscr{X}$ denote the system state at time t and $u_t \in \mathscr{U}$ the action applied by the agent. A policy defines how actions are selected based on the available state information. In the general case, a stochastic policy is defined as

$$\pi(u|x) = \Pr(U_t = u|X_t = x), \tag{20}$$

while a deterministic policy directly maps states to actions according to $u_t = \pi(x_t)$. The selected action is applied to the environment, which then

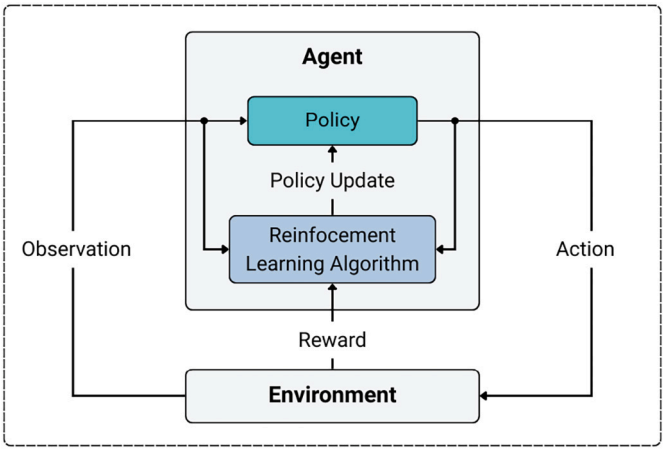

Fig. 8. Reinforcement learning agent-environment interaction loop.

transitions to a new state and produces a reward signal.

The reward signal provides immediate feedback on the desirability of the action. While it defines the optimisation objective, it does not directly prescribe how to achieve it. Instead, the objective is to maximise the expected cumulative discounted return

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}, \tag{21}$$

where $\gamma \in [0,1)$is the discount factor that determines the relative importance of future rewards (or, equivalently, future costs).

Value functions quantify the expected long-term performance of a control policy. The state-value function is defined as

$$V^{\pi}(x) = \mathbb{E}_{\pi}[G_t | X_t = x], \tag{22}$$

while the action-value function evaluates state-control pairs:

$$Q^{\pi}(x,u) = \mathbb{E}_{\pi}[G_t | X_t = x, U_t = u]. \tag{23}$$

Unlike the instantaneous reward, value functions capture the long-term consequences of control decisions. This allows the agent (or controller) to prefer actions (control inputs) that may be locally sub-optimal but lead to improved closed-loop performance over time. Although value functions are not shown explicitly in the policy block of Fig. 8, they are typically learned as part of the RL algorithm and are used either to construct improved control laws or to guide policy updates.

In some RL settings, the agent may possess or learn a model of the system dynamics. Such a model characterises the system evolution and reward generation, typically through the transition density $p(x_{t+1}|x_t, u_t)$ and the reward function $r(x,u)$.

Model-free methods learn control policies or value functions directly from interaction data without explicitly identifying the system dynamics. Examples include value-based methods such as Q-learning and policy-gradient approaches such as Proximal Policy Optimisation (PPO). In contrast, model-based RL methods rely on an explicit system model to predict future trajectories and rewards. This model can be used for planning, policy evaluation, or the generation of synthetic data, often improving data efficiency and safety.

This agent-environment framework provides a general and unifying abstraction for a wide range of learning and control problems. It serves as the conceptual foundation for more specialised formulations, including multi-armed bandit problems and Markov Decision Processes, which are discussed in the following subsections.

2.4.2. Multi-arm bandits

Multi-armed bandit problems represent a simplified class of RL tasks in which the agent repeatedly selects from a finite set of actions (control inputs) and receives a reward associated with each selection. In contrast to general RL problems, multi-armed bandits do not involve state transitions or long-term planning. Each action choice affects only the immediate reward and does not influence future decision contexts.

The central challenge in multi-armed bandit problems is the trade-off between exploration and exploitation. Exploration involves selecting actions with uncertain outcomes in order to improve knowledge about their reward distributions, while exploitation focuses on choosing actions known to yield high rewards based on past experience. Despite their simplicity, multi-armed bandits capture essential aspects of decision-making under uncertainty and form the foundation for many RL algorithms. They are particularly relevant in applications where decisions must be made sequentially without explicit state dynamics, such as parameter tuning or adaptive control under static operating conditions.

2.4.3. Markov decision process

More general RL problems are commonly formalised using the framework of Markov Decision Processes (MDPs). An MDP provides a mathematical model for sequential decision-making in environments where outcomes depend on both the agent's actions and stochastic dynamics. An MDP is defined by the tuple

$$\mathscr{M} = (\mathrm{X}, \mathrm{U}, p, r, \gamma), \tag{24}$$

where $\mathscr{X}$is the state space, $\mathscr{U}$ the control input space, $p(x'|x,u)$ the state transition probability density, $r(x,u)$the reward function, and $\gamma \in [0,1)$the discount factor.

The defining characteristic of an MDP is the Markov property, which assumes that the future evolution of the environment depends only on the current state and control input, and not on the full history of past interactions. Under this assumption, the environment's dynamics can be fully characterised by the one-step transition model

$$p(\mathrm{x}', r|\mathrm{x}, \mathrm{u}) = \Pr(\mathrm{X}_{t+1} = \mathrm{x}', R_{t+1} = r | \mathrm{X}_t = \mathrm{x}, \mathrm{U}_t = \mathrm{u}), \tag{25}$$

which specifies the probability of transitioning to state $x'$and receiving reward r after applying control input u in state x.Within this framework, a policy specifies the agent's behaviour, and value functions quantify the expected return from states or state-action pairs based on the transition dynamics and reward structure. For a given policy π, the state-value function satisfies the Bellman expectation equation

$$V^{\pi}(x) = \sum_{u} \pi(ux) \sum_{x',r} p(x', r|x, u)[r + \gamma V^{\pi}(x')], \tag{26}$$

which expresses the recursive relationship between the value of a state and the values of its successor states. This equation forms the theoretical foundation of many RL algorithms, enabling policy evaluation and serving as the basis for iterative policy improvement.

Many RL algorithms can be interpreted as methods for solving MDPs, either by estimating value functions, improving policies, or jointly performing both steps. Multi-armed bandit problems can be viewed as a special case of MDPs with a single state and trivial transition dynamics. Consequently, the MDP formulation provides a unifying framework that encompasses both simple and complex RL problems, enabling systematic analysis and algorithmic development across a wide range of applications.

2.4.4. Q learning

Q-learning is the simplest RL algorithm. Although RL conventionally denotes actions by a and states by s, to maintain consistency with the dynamic-systems notation introduced above, we shall instead use u for the control input and x for the system state.

The initialisation point for RL begins by storing a table of the Q-values for each action and state pair, defined by $Q(x,u)$, which represents the expected cumulative reward when action u is taken in state x. In tabular Q-learning, x is discrete: in a multi-dimensional system, one

might encode each state as a vector of discrete levels (for example, [0,0] → state 0, [0,1] → state 1, [1,0] → state 2, [1,1] → state 3, and so on). Initially, every entry in the Q-table is set to zero (or to small random values), the learning proceeds in episodes of fixed duration. For example, in a dynamic system with a sample rate of dt, the episode can be considered by a temporal window equal to $T = N{\cdot}dt$, and therefore it can end after a pre-defined N interactions between the agent and the system. However, other condition-based endings of the episode can be used. That said, at each time step:

1. The agent selects an action according to an exploratory strategy, commonly ε-greedy, whereby with probability ε a random action is chosen. The value of ε is linearly reduced moving on during the training, changing from exploration to the policy exploitation.
2. The agent observes the immediate reward r_t (function of x,u) and the next state x_t coming from the environment, as results of the applied action.
3. Then the agent updates the corresponding table entry via the rule:

$$Q(x_t,u_t) \leftarrow Q(x_t,u_t) + \alpha[r_t + \gamma \max Q(x_{t+1},u) - Q(x_t,u_t)]. \tag{27}$$

Where α is the learning rate and γ the discount factor and $\max Q(x_{t+1},u)$ is the estimated value of the best action at the next state. Q-learning is both value-based, since its policy simply selects the action with the highest Q-value, and off-policy, because it can learn from experiences generated by any behaviour policy. This means that, the agent can store each transition (x_t,u_t,x_{t+1},r_t) in an experience replay buffer and sample random mini-batches for updates, breaking temporal correlations, improving data efficiency and accelerating convergence towards the optimal policy [95]. Despite its simplicity, tabular Q-learning does not scale well to continuous, high-dimensional state or action spaces without discretisation or other approximations. Nevertheless, for complex AM tasks characterised by continuous control variables and intricate process signatures, one may exploit the generalisation capabilities of neural networks via Deep Q-Learning (DQN). In DQN, the action-value function is no longer stored in a discrete lookup table but is instead approximated by a neural network, thereby accommodating high-dimensional or continuous state spaces [96]. The treatment of control actions remains essentially the same: the action domain is discretised, each level is assigned an index, and when multiple control variables are involved, the previously described encoding scheme is applied.

2.4.5. Deep Q learning

As discussed for standard tabular Q-learning the update does not use the action actually taken at the next step. Instead choosing the one with the highest estimated value, standard Q learning bootstraps on the greedy action during the exploration, updating the Q-values regardless of what the agent does. In a DQN algorithm, such as SARSA, the weights of the neural network (θ) that estimate possible the Q-values based on the states are still updated based on the temporal error, but instead of using the target as the maximum over all next actions ($\delta_{TD} = \max Q(x',u') - Q(x,u)$), use the actual next action, making it an on-policy algorithm. SARSA uses Eq. (28) to compute the error, while using Eq. (29) for backpropagation to update the weights, considering the gradient of the loss $L = \frac{1}{N}\sum \delta_{TD}^2$.

$$\delta_{TD} = r_t + \gamma Q(x_{t+1},u_{t+1}) - Q(x_t,u_t) \tag{28}$$

$$\theta \leftarrow \theta + \alpha \delta_{TD} \nabla_\theta Q(x_t,u_t,\theta) \tag{29}$$

The application of DQN in manufacturing has been widely recognised for its ability to address numerous decision-making challenges. Whilst a comprehensive survey of DQN across all manufacturing domains lies beyond the remit of this review, we briefly highlight a couple of studies that demonstrate its effective implementation in systems with continuous state spaces, such as sensor readings in AM processes, and multi-dimensional action spaces. These works offer valuable insights into policy-approximation techniques that inform the examples discussed below. One of the most studied applications of DQN is the scheduling of manufacturing systems. For example, Zhao et al. [97] proposed a DQN agent that observes a five-dimensional continuous state, average machine utilisation, its standard deviation, average job completion rate, average remaining-process waiting rate and overall load (all normalised to [0,1]), and at each decision step selects one of ten classic dispatching rules. A simple reward (estimated average slack minus estimated average remaining processing time) encourages the agent to maximise slack and minimise remaining work, enabling it to learn a dynamic policy that minimises total job tardiness in a randomly arriving, uncertain job-shop environment. Another example is the work of Vespoli et al. [98], that used a DQN as high-level controller for the Constant Work In Progress (CONWIP) on a flow shop line, defining the state vector of per-machine and queue statistics (sample means and standard deviations of processing times), the target throughput, normalised throughput error and total WIP (vector of length equal to 2*M + 3), and the control action as discrete choice of allowable CONWIP levels (from 1 up to a maximum derived from the rearranged CONWIP throughput equation). In their study, the authors proposed a smooth, Gaussian-shaped function of the normalised absolute error between actual and target throughput, penalising large deviations while guiding the agent gently towards zero error, enabling to maintain the production throughput on the line also in variable and uncertainties condition typical of flexible Industry 4.,0 context.

In both examples, RL serves as a powerful mechanism for adaptive, high-level control in complex scheduling environments. In the first study, a DQN agent dynamically selects among ten canonical dispatching rules, while in the second, an RL controller autonomously determines the optimal quantity of WIP to release at the first machine, accounting for downstream effects across the entire line. Together, these applications exemplify how RL enables self-tuning "expert systems" that translate high-level objectives into finely-graded operational decisions, making optimal actions without know upfront what is the best one. Moreover, multi-agents DQN have been employed for finding optimal fixture plans for components during drilling tasks [99].

2.4.6. Policy gradient algorithms

Despite their effectiveness, Q-learning methods scale poorly as the action space grows. For example, three binary control parameters already yield $2^3 = 8$ possible actions, while three parameters with three levels each give $3^3 = 27$. This combinatorial explosion makes discrete Q-networks impractical for truly continuous control problems. Policy gradient algorithms sidestep this "curse of dimensionality" by parameterising the policy directly, rather than learning a Q-value for every action, so that given a state, the network outputs a continuous action (or a distribution over actions) in a single forward pass. Depending on the formulation, they may employ a stochastic policy (e.g. sampling from a learned distribution) or a deterministic policy (directly regressing the optimal action).

In algorithms such as SARSA and DQN, a neural network is employed to approximate the value function. Specifically, in SARSA the network estimates the state-action value function $Q(x,u) \approx Q_\theta(x,u)$, where θ denotes the parameters of the network, with the aim of maximising the expected return. Similarly, in DQN a neural network approximates the optimal action-value function $Q^*(x,u)$, and the greedy policy (π) is derived as in $\pi(x) = \arg\max_x Q_\theta(x,u)$.While these methods are effective in environments with moderate state and action spaces, they become intractable when the action space is very large or continuous, in which discretisation of continuous actions is computationally expensive and introduces approximation errors [100]. To address this issue, gradient-based policy optimisation methods are introduced. Instead of estimating a value function and deriving a policy indirectly, these methods directly parameterise the policy as $\pi_\theta(u|x)$, where the objective is to

maximise the expected return. A fundamental example is the REINFORCE algorithm, which applies the policy gradient theorem. The gradient of the performance objective is expressed as in Eq. (30), supposing a stochastic policy:

$$\nabla_\theta J(\pi_\theta) = \sum_{t=0}^{T} R_t \nabla_\theta \log \pi_\theta(u|x). \tag{30}$$

Then the parameters are updated by using gradient ascent $\theta = \theta + \alpha \nabla_\theta J(\pi_\theta)$. Since in REINFORCE the policy is stochastic, for continuous action spaces, the control policy $\pi_\theta(u|x)$ is often modelled as a Gaussian distribution. This means that, in the simplest case, the policy network outputs the mean μ_θ and (diagonal) covariance Σ_θ so that the action is sampled from the distribution in Eq. (31), enabling smooth exploration in continuous domains. This formulation allows policy gradient methods to overcome the limitations of value-based approaches in high-dimensional or continuous action settings:

$$u = \pi_\theta(u|x) \sim N(x|\mu_\theta(x), \Sigma_\theta). \tag{31}$$

From a design perspective, the output of the network is equal to $2{\cdot}size(u)$, one for the mean and another one for the variance of the Gaussian distribution associated with each action. However, despite their high flexibility, gradient-based methods have a huge drawback associated with the high variance associated with Monte Carlo return estimates caused by the calculation of rewards each time step and training instability [101]. A common way to reduce variance is to subtract a baseline from the rewards in the gradient in Eq. (32) that does not depend on the action taken from the policy. In this way, it does not introduce any bias to the policy gradient. This could be a random number, but a good choice is to use the Q or V values as a baseline, as reported in Eq. (32), leading to an Actor-Critic DRL architecture

$$J(\pi_\theta) = \sum_{t=0}^{T} (R_t - Q(x,u)) \nabla_\theta \log \pi_\theta(u|x). \tag{32}$$

If the Q function is approximated by a neural network, such in the case of DQN or SARSA, the structure of the agent becomes an Actor (gradient-based) plus a Critic (baseline approximation). The Critic, is responsible for learning a value or Q function, trained by minimising the temporal-difference (TD) error, while the second network, known as the Actor, parameterises the policy π_θ and it is optimised via gradient ascent on the policy objective. This dual-network structure allows the critic to provide a lower-variance estimate of action values, thereby stabilising the actor's policy updates and improving learning efficiency. Prominent examples of Actor–Critic algorithms include the Deep Deterministic Policy Gradient (DDPG) method, which is particularly suited to continuous action spaces and employs a deterministic actor combined with a critic trained using TD learning. Another widely adopted method is Proximal Policy Optimisation (PPO), which incorporates a clipped surrogate objective to prevent excessively large policy updates, thereby achieving a balance between learning stability and exploration.

In the case of DDPG, the learning process is split between two networks with distinct objectives. The critic is trained by minimising the TD error, i.e. the mean squared error between the predicted action–value function and a bootstrapped target. The actor, on the other hand, implements a deterministic policy, meaning that for a given state it directly outputs $(\pi_\theta(x))$ an action rather than sampling from a stochastic distribution. Its parameters are optimised to maximise the critic's estimate of the action-value function by applying gradient ascent. In contrast, PPO represents an advanced approach to stochastic Actor-Critic methods. In contrast, PPO is based on trust-region learning (TRL), proposed by Schulman et al. [102], which improves the stability of the training by restricting the policy update size at each learning iteration, preserving monotonic improvement guarantees using a second-order optimisation algorithm based on the constraint reported in Eq. (33).

$$D_{KL}(\pi_{\theta'}(u,x); \pi_\theta(u,x)) \leq \delta \tag{33}$$

Where θ' are the parameters of the new policy and θ of the old policy, D_{KL} refers to the Kullback-Leibler (DKL) divergence between the two stochastic policies. TRPO addresses policy optimisation with hard constraints, but this becomes computationally cumbersome in high-dimensional spaces such as neural networks. To overcome this, the same authors introduced PPO [103], which replaces the hard constraint with a clipping objective or KL-divergence regularisation. This modification reduces computational complexity while preserving the essential properties of the TRPO framework, as argued through a "proof by analogy". In particular, the concept of the advantage function is introduced, as defined in Eq. (34), which depends on the KL-divergence to prevent excessively large policy updates during training.

$$A_t(x,u) = Q(x,u) - \beta KL[\pi_{old}(x_t), \pi_{new}(x_t)] \tag{34}$$

Both Policy Gradient and PPO algorithms are on-policy algorithms, which means that they collect data by interacting with the environment under the current policy and then update it using those data. For that reason, on-policy algorithms possess poor sample efficiency, since a large number of interactions are needed to converge. On the other hand, off-policy algorithms, such as DDPG which sample from a reply buffer, decouple data collection and policy updates, giving the ability to use also past data during training. To further improve the efficiency of algorithms such as PPO and DDPG, more advanced methods have been developed. Soft Actor-Critic (SAC) [104] extends the actor-critic framework by introducing an entropy term into the objective, which encourages more exploratory policies and enhances robustness in continuous action spaces. Twin Delayed DDPG (TD3) [105] addresses instability and overestimation issues in DDPG by employing two critics, delaying actor updates, and adding target policy smoothing, thereby achieving more stable and reliable learning.

The applications of DDPG and PPO in industry are numerous, particularly in machine-level manufacturing processes, where they serve as data-driven feedback controllers well suited to the continuous nature of such operations. An example is the work of Zhang et al. [106], that used PPO to design a constant-force grinding controller, enabling the system to maintain stable grinding force despite uncertainties and disturbances with reduced grinding force fluctuation compared to traditional control methods. Another example is the work of Hao et al. [107], in which the authors designed a DDPG to adaptive change based on the reference input the optimal gains for a PID controller of a hydraulic servo systems of injection moulding machines, achieving higher tracking accuracy, faster convergence, and improved robustness compared with conventional methods. Moreover, these methods have been applied with success also in the field of manufacturing systems. Indeed, in Zhang et al. [108], employed a PPO-based scheduler for a flexible job shop environment. The state was constructed by concatenating the current operation's features (operation type, nominal processing time, and normalised slack) with each machine's attributes (type, buffer occupancy, and cumulative workload). At each decision point, the scheduler selected one discrete action, encoded as a one-hot vector of length m + 1 m + 1 m + 1 (with mmm denoting the number of machines): either 'standby' or assignment to machine 1…m.

This illustrates how algorithms such as PPO and DDPG can be effectively applied across diverse areas of manufacturing systems and machinery, demonstrating their versatility in handling not only continuous but also discrete action spaces, as shown in the examples. Furthermore, recent studies have explored their integration into additive manufacturing, where their potential is increasingly recognised. The details of these applications will be discussed in the following section.

2.5. Reinforcement learning in the context of control theory

In traditional control theory, the design of a feedback controller relies on a limited set of fundamental elements. These include:

- the *plant*, which represents the model of the process and constitutes the basis for controller design;
- the *controller*, defined through a specific control law and associated cost function weights;
- the *state*, representing the variables to be regulated or observed; and
- the *control action*, which acts on the process to achieve the desired behaviour.

The same conceptual structure applies to RL.

- The notion of *state* (or observation) directly corresponds to the state in classical control. In its simplest formulation, the RL state may contain the same variables used by a traditional controller. However, one of the key advantages of RL is that the state can be augmented with additional information that is not necessarily subject to regulation but may still be useful for decision-making. This flexibility enables the inclusion of contextual, historical, or auxiliary variables that are difficult to incorporate into classical control formulations.
- Similarly, the concept of *environment* in RL plays the same role as the plant in control theory. The environment may be represented by any model suitable for control design, ranging from simple linear dynamics or step-response models to more complex non-linear formulations, such as Hammerstein or Hammerstein-Wiener [109] structures. Importantly, these modelling choices are not specific to RL, but are equally employed in advanced control strategies, including non-linear model predictive control. As such, the environment-plant correspondence is direct and conceptually equivalent.
- With respect to performance optimisation, classical control relies on cost functions, typically quadratic, whose weights determine the trade-off between tracking accuracy, control effort, and stability. In RL, this role is fulfilled by the reward function. While reward functions often include terms equivalent to tracking error and control effort penalties, RL allows greater flexibility in incorporating additional objectives. These may include process efficiency, operational stability, or quality-related indicators that are not strictly dynamical in nature. Unlike classical cost functions, the reward does not need to adhere to a specific analytical structure, provided it is well defined and bounded.

From this perspective RL can be interpreted as a data-driven extension of control theory, where the policy replaces the explicit analytical controller design. Model-free reinforcement learning algorithms, which are predominantly adopted in feedback control applications, do not eliminate the need for a model, but instead shift the modelling effort to the environment representation used during training. This is directly analogous to controller design based on a plant model in classical control. Consequently, RL should not be viewed as an alternative to control theory, but rather as a natural extension that becomes particularly valuable when classical control formulations become difficult to apply due to strong non-linearities, complex coupling, or heterogeneous data sources. As discussed in the following sections, this correspondence also extends to hardware and implementation considerations, where the same computational architectures used for advanced control (e.g. MPC) are often suitable for RL-based controllers, with additional requirements dictated primarily by data type and inference complexity.

3. Advances of reinforcement learning in additive manufacturing

To establish the scope of this review, a systematic search was conducted in Scopus using the query “reinforcement learning” AND “additive manufacturing”. After filtering out works unrelated to manufacturing processes, such as generic scheduling studies or machine learning approaches without RL components, 43 papers published between 2018 and 2025 were identified. Given the diversity of applications and the process-dependent nature of the challenges, the reviewed works are organised into three subsections: Powder Bed Fusion (PBF), Direct Energy Deposition (DED), and Fused Deposition Modelling (FDM). PBF represents the area where RL has been predominantly applied to melt pool monitoring, defect suppression, and design or toolpath optimisation. Research in DED has focused on path planning, process parameter optimisation, and defect control, particularly in wire arc and laser-based systems. Finally, FDM covers extrusion-based processes, together with related polymer AM techniques, where RL has been employed for toolpath planning, extrusion and thermal regulation, defect mitigation, and the emerging field of multi-material and 4D printing. This classification provides a structured basis for comparing methods, objectives, and outcomes across different AM processes.

3.1. Powder bed fusion

3.1.1. Applications of reinforcement learning

Recent advances in RL have been increasingly applied to enhance process monitoring, control, and design in laser-based PBF. Wasmer et al. [110] introduced a novel in situ quality monitoring approach that combines acoustic emission (AE) sensing with RL classification. In this work, the authors proposed a Q-learning-based classifier for each class, enabling hard classification without relying on supervised classification mapping. Unlike conventional melt pool imaging or thermal monitoring, AE enables subsurface defect detection. Moreover the trained RL classifier achieved 74–82% accuracy in distinguishing porosity levels in 316L stainless steel, demonstrating the potential of RL-driven sensing for real-time quality assurance.

Beyond direct process monitoring, RL has also been integrated into generative design and toolpath optimisation. Venugopal and Anand [111] proposed a generative RL framework for topology optimisation that combined an Upper Confidence Bound (UCB) search strategy with Design for AM filters, producing diverse, manufacturable designs that outperformed traditional SIMP-based approaches in compliance and thermal resistance. Toolpath optimisation has been explored by Huang et al. [112], who developed a DQN-based planner that generates paths directly on graph representations of target geometries. Their approach reduced deformation, sharp turns, and thermal distortion in both wireframe fabrication and LPBF, outperforming conventional heuristic strategies. Fig. 9 illustrates how the RL-based planner significantly reduces sharp turns compared with DFS-generated paths, thereby improving fibre adhesion. Qin et al. [113] extended this direction by proposing a deep RL-based framework for adaptive toolpath generation to reduce residual stresses, achieving up to 47% distortion reduction compared to Zigzag patterns in thin-plate experiments, and outperforming Chessboard and adaptive strategies in both simulation and validation tests.

Efforts have also been made to extend RL-based control beyond single-layer stabilisation. Vagenas et al. [114] introduced an Adaptive Weighted Actor-Critic (AWAC) method for multi-layer SLM thermal management, outperforming traditional PID control and previous RL algorithms in maintaining melt pool stability across full builds. Park et al. [115] further advanced real-time melt pool homogenisation by combining predictive scan-path information with reactive melt pool feedback in a geometry-informed RL framework. Their controller achieved over 30% error reduction and reduced melt pool variation in experimental trials, delivering performance comparable to feedforward+PID control but at significantly lower optimisation cost. Faizan Mohamed et al. [116] coupled Q-learning with a physics-informed porosity model to optimise process parameters in Al alloy LPBF, achieving 99.5% relative density and demonstrating the potential of embedding physics-based insights into RL for defect mitigation. Similarly, Yin et al. [117] integrated Q-learning with an analytical heat conduction model to dynamically tune laser power and scan velocity for molten pool stabilisation. Validated through simulations and Ti-6Al-4V experiments, their framework-maintained target melt pool depth and

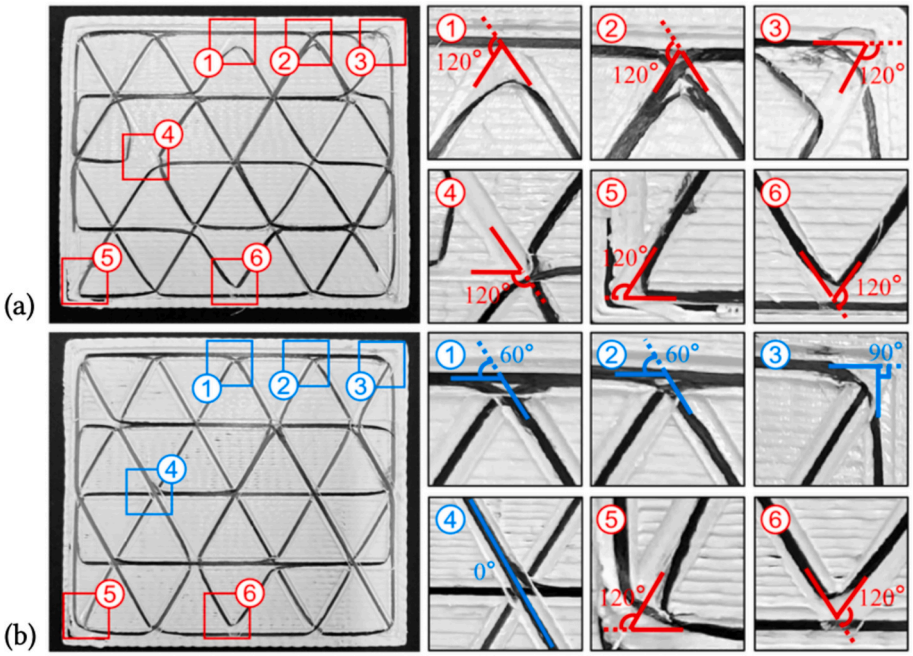

Fig. 9. Comparison of toolpaths generated for carbon fibre reinforced (CCF) printing: (a) unoptimised depth-first search (DFS) path with six sharp turns (red squares, adhesion failures), and (b) reinforcement learning–based planner path with only two sharp two and improved fibre adhesion (blue squares show adjusted angles). Adapted from Huang et al. [112]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

mitigated boundary overheating during spiral scan paths, thereby improving dimensional accuracy and process reliability.

These results highlight the potential of combining RL-informed design principles with PBF manufacturing for biomedical applications such as scaffolds.

3.1.2. Application of RL in feedback control

In the domain of process control, Knaak et al. [118] integrated HDR imaging, CNN-based roughness classification, and Q-learning for inter-layer quality optimisation in LPBF. In particular, they define discrete actions such as reducing, increasing, or maintaining the laser power and scan speed by a fixed quantity, while the state consists of the control action taken, the output of a CNN classifier, and an estimate of surface roughness, thereby demonstrating how an additional monitoring module can be effectively incorporated into the control action decision process. Their framework improved classification accuracy by 12% compared with low dynamic range imaging, identified parameters that reduced average roughness to 3.38 μm, and converged faster than Q-learning while avoiding defective surfaces.

Malik et al. [119] advanced this direction with a digital twin framework combining finite element simulations, recurrent neural networks, and a PPO agent to suppress lack-of-fusion defects in stainless steel. The method accurately predicted defect likelihood and adaptively adjusted parameters, with experimental validation confirming its potential for real-time corrective control. Building on PPO-driven strategies, Ogoke and Barati Farimani [120] trained a PPO agent on continuum-scale heat conduction simulations to regulate thermal behaviour. The agent adjusted laser velocity or power to reduce melt pool depth errors by 64–91% compared to constant-parameter baselines, demonstrating the promise of DRL for adaptive defect suppression. However, this study lacks a sim-to-real transfer, leaving open the question of whether thermal images can be effectively used as input for process control of laser speed, given the difficulties associated with online implementation on actual machines. Grais et al. [121] further showed in simulation that MIMO PPO controllers could simultaneously regulate laser power and scan velocity, yielding more stable melt pools with reduced control effort relative to single-parameter control. Complementing these approaches, Vagenas and Panoutsos [122] investigated stability in PPO-based thermal control through simulation, showing that heuristic reward shaping was insufficient and highlighting the need for Lyapunov-based guarantees to ensure robust deployment. Table 1 summarises the reviewed works, outlining their methods, objectives and outcomes.

To complement the high-level overview in Table 1, Table I in Appendix A provides a formulation-level comparison of the PBF studies, documenting the underlying RL formulation and the definitions of states, actions, rewards, and policy or value-function architectures, including how these elements are constructed from sensing, simulation, or geometric information.

3.1.3. Comparison analysis

Table 1 and Table I in Appendix A reveal that RL has been predominantly applied in PBF to problems of thermal regulation and melt pool control, reflecting the central role of thermal stability in ensuring part quality. Across these studies, the RL objective is typically defined as tracking a target melt pool depth, width, or temperature, with states constructed from thermal observations ranging from high-dimensional temperature fields to compact, low-dimensional summaries such as average layer temperature or discretised melt pool error. The corresponding action spaces most commonly involve continuous or discretised adjustments of laser power and scan velocity, applied at scan-point, timestep, or layer resolution. Reward functions are consistently formulated as shaped tracking objectives, penalising deviation from reference values and often including regularisation terms to suppress excessive thermal fluctuations. Within this application class, policy-gradient methods, particularly PPO are the most frequently adopted algorithms. As shown in Table 1 and Table I in Appendix A, PPO-based controllers demonstrate substantial reductions in melt pool tracking error and improved stability relative to heuristic or PID-based baselines. Several studies further extend these formulations to MIMO control, simultaneously regulating laser power and scan velocity, and report improved accuracy and reward consistency compared with single-parameter control strategies.

A second prominent research direction is defect detection and

Table 1
Summary of reinforcement learning applications in PBF, including methods, optimisation objectives, and reported outcomes.

Author (Year)	RL/ Optimisation Method	Objective	Key Outcome
Wasmer et al. [110] (2018)	Q-learning (ε-greedy, tabu search)	AE-based porosity monitoring	Achieved 74-82% accuracy in defect detection
Knaak et al. [118] (2021)	Model-based RL + CNNs	Inter-layer quality optimisation via HDR imaging (surface roughness, distortion)	Achieved a 12% accuracy gain compared with LDR imaging; average roughness of 3.38 μm; faster than Q-learning; avoided more than 10% defective surfaces
Malik et al. [119] (2024)	PPO + RNN	Suppression of lack-of-fusion defects	Accurate defect prediction; adaptive parameter control; validated on 316L SS
Ogoke and Barati Farimani [120] (2021)	PPO	Thermal regulation via laser velocity/ power adjustment	Reduced melt pool depth errors by 64-91%; enabled adaptive defect suppression
Grais et al. [121] (2023)	PPO (MIMO)	Simultaneous control of laser power and velocity	Achieved stable melt pool depth and width; improved accuracy and rewards compared with single-parameter control
Vagenas and Panoutsos [122] (2023)	PPO	Stability analysis of thermal control policies	Showed that heuristic reward shaping was insufficient; motivated Lyapunov-based guarantees for robust deployment
Venugopal and Anand [111] (2023)	UCB + DfAM filters	Generative design via topology optimisation (compliance & thermal)	Produced diverse, manufacturable designs; outperformed SIMP in compliance & thermal resistance
Huang et al. [112] (2024)	DQN	Graph-based, on-the-fly toolpath planning	Reduced deformation, sharp turns, and thermal distortion vs. heuristics
Qin et al. [113] (2024)	Deep RL	Adaptive toolpath generation for stress reduction	Reduced distortion by up to 47% compared with Zigzag; outperformed Chessboard and ATG strategies
Vagenas et al. [114] (2024)	AWAC	Multi-layer thermal management	Outperformed PID and prior RL methods; improved melt pool stability with more stable training and reduced variance
Park et al. [115] (2025)	Q-learning (geometry-informed)	Real-time melt pool homogenisation / regulation (laser power control)	Reduced error by more than 30%; reduced melt pool variation; comparable to feedforward+PID at about 20 times lower optimisation cost
Faizan Mohamed et al. [116] (2025)	Q-learning + physics-informed porosity model	Process parameter optimisation for porosity reduction	Achieved more than 99.5% relative density; demonstrated effectiveness of embedding physics-based insights into RL
Yin et al. [117] (2025)	Q-learning + analytical heat conduction model	Dynamic tuning of laser power & velocity for molten pool stabilisation	Maintained target melt pool depth; mitigated boundary overheating; improved dimensional accuracy

suppression, including porosity monitoring, lack-of-fusion mitigation, and surface quality optimisation. In these studies, RL is commonly integrated with physics-based models or learned defect predictors, which serve either as environment models or as components of the reward function. State representations range from discretised process parameter grids to features extracted from acoustic, optical, or thermal sensing. Actions correspond to incremental parameter updates, while rewards are designed to maximise relative density, minimise defect likelihood, or maintain defect indicators within acceptable bounds. The reported outcomes indicate that incorporating physics-informed or model-assisted feedback improves both learning efficiency and achievable quality metrics.

Beyond real-time process control, several studies apply RL to toolpath planning and design optimisation problems. In these cases, the RL state is primarily geometric or graph-based, encoding local connectivity, scan history, or laser position, while thermal effects are captured indirectly through reward shaping or internal simulation. Value-based deep RL methods, particularly DQN variants, consistently outperform heuristic scan strategies, achieving notable reductions in distortion and thermal accumulation. Related work on generative and topology optimisation formulates the problem as a bandit or value-based learning task embedded within SIMP-style frameworks, enabling exploration-driven design diversity while respecting additive manufacturing constraints.

Taken together, the formulation-level comparison in Table 1 highlights a clear trade-off between model fidelity and implementation complexity. High-dimensional sensory states and deep neural policies enable fine-grained control but raise challenges related to stability and computational cost. Conversely, several recent studies demonstrate that low-dimensional, discretised state spaces combined with tabular Q-learning and physics-based environment models can achieve competitive performance with significantly reduced optimisation overhead. These findings suggest that the effectiveness of RL in PBF depends less on algorithmic complexity than on the careful alignment of state, action, and reward definitions with process physics and sensing constraints.

3.2. Direct energy deposition

3.2.1. Applications of reinforcement learning

RL has also been applied to advance control, monitoring, and path planning in DED processes, particularly in wire arc and laser-based systems. Dharmadhikari et al. [123] introduced a Q-learning framework for process parameter optimisation in Laser-Directed Energy Deposition (L-DED). By modelling the laser as an RL agent in a digital twin environment based on the Eagar-Tsai formulation, the approach identified optimal power-velocity combinations that maintained target melt pool depth. Experimental validation on SS316L single tracks confirmed predictions within 50 μm accuracy, and hyperparameter studies demonstrated the robustness of the method under scarce-data conditions. This work highlights RL's potential for efficient, model-free calibration of metal AM processes, especially when prior process knowledge is limited.

Beyond parameter tuning, a central challenge in DED lies in efficient and reliable path planning. Petrik and Bambach contributed two frameworks for WAAM: RLTube [124], which reduced layer counts in thin-walled tubular builds by up to 30% through RL-based optimisation (simulation-based), and RLPlanner [125], a PPO-driven system that jointly determines deposition paths and process parameters. Experimental validation on benchmark parts showed RLPlanner achieved geometric deviation of less than 0.5 mm while reducing the need for manual intervention, underscoring RL's promise for geometry-adaptive, automated planning. Building on this, Sideris et al. [126] integrated PPO with Monte Carlo Tree Search and reduced-order thermal modelling, producing deposition paths that balanced productivity and thermal uniformity, reducing temperature deviations by nearly 20% in simulation. Experimental results further confirmed its effectiveness, as shown in Fig. 10, where RL-informed continuous strategies reduced

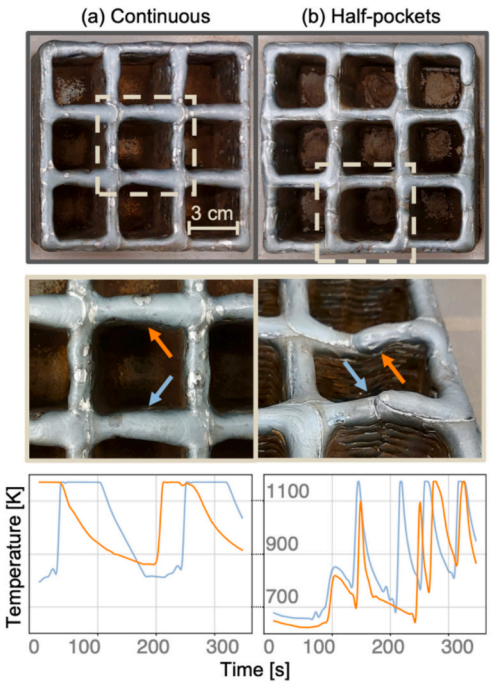

Fig. 10. Experimental parts produced using reinforcement learning–informed path strategies from the Pareto front: continuous (left) and half-pocket segmented (right). Continuous deposition exhibited overheating-induced defects, while half-pocket segmentation resulted in welding defect accumulation from frequent starts and stops. Adapted from Sideris et al. [126].

overheating compared to segmented ones, while maintaining build integrity and efficiency. Ferreira et al. [127] applied Multi-Armed Bandit algorithms with Thompson sampling to optimise WAAM path strategies, achieving more efficient trajectories. In a related study, Ferreira et al. [128] proposed the Advanced-Pixel method, which delivered faster convergence, shorter build times, and improved scalability to complex lattice-type geometries. Together, these studies highlight RL's capacity to address the inherently multi-objective nature of DED path planning, where thermal stability, deposition efficiency, and geometric accuracy must be jointly optimised.

Finally, RL has been explored for quality inspection and adaptive adjustment in DED variants. Li et al. [129] proposed a multiclass reinforced active learning framework for inkjet-based deposition, integrating graph convolutional networks with a deep Q-network to classify droplet pinch-off behaviours. By reducing labelling requirements to around 18% of conventional methods, the framework improved classification accuracy whilst also supporting parameter adjustment for mode control. This study demonstrates RL's potential not only for process optimisation, but also for intelligent quality monitoring in DED.

3.2.2. *Application of RL in feedback control*

Dharmawan et al. [130] proposed a model-based RL framework for in situ corrective control of robotic Wire Arc Additive Manufacturing (WAAM). By employing Gaussian Process Regression to capture the process dynamics, each deposition point acted as an RL agent based on Q learning capable of adjusting parameters such as torch speed and wire feed rate, with the aim of maintaining a constant layer height. Experiments on bronze and stainless-steel builds demonstrated improved geometric uniformity and reduced error accumulation, illustrating RL's potential for real-time adaptive control in multi-layer deposition. However, the applicability in online feedback control has not yet been explored, and the approach is currently limited to acting as an intra-layer parameter adjustment.

Similarly, Mattera et al. [15] developed a WAAM-specific simulator that combines reduced-order modelling with DDPG control, enabling near real-time MIMO control of bead geometry in simulation. In this work, a novel reward function has been developed, with potential applicability in RL frameworks that incorporate additional indicators such as energy efficiency and feedback quality from monitoring system modules. While challenges remain for industrial deployment, the study illustrates how deep RL can be integrated with data-driven modelling for adaptive process control. Table 2 summarises the reviewed works, highlighting their methods, objectives, and outcomes.

Table 2 summarises RL applications in directed energy deposition at a high level, emphasising the methods employed, optimisation objectives, and reported outcomes. However, such summaries do not convey how RL is operationalised within each study. To address this, Table II in Appendix B examines the reviewed works from a formulation-level perspective, detailing the construction of state representations, action spaces, reward functions, and learning architectures. This comparison exposes differences in modelling assumptions, degrees of observability, and control scope, as well as the roles of sensing, simulation, and process models within the learning loop.

3.2.3. *Comparison analysis*

Table 2 and Table II in Appendix B a comparative overview of RL applications in DED, with a particular emphasis on feedback control, deposition path planning, and geometric regulation. Compared with PBF, the reviewed DED studies exhibit a broader diversity of control objectives, RL formulations, and learning paradigms, reflecting the increased process flexibility and sensing accessibility typical of DED systems.

A central group of studies focuses on process parameter control for geometric consistency, including melt pool depth, bead geometry, droplet volume, and layer height regulation. In these works, the RL state is generally defined using low-dimensional geometric or process-related measurements, such as bead width and height, droplet volume error, or local print height derived from in situ sensing or reduced-order models. Actions correspond to continuous or discretised updates of deposition parameters, including laser or arc power, torch or welding speed, wire feed rate, and waveform characteristics. Reward functions are predominantly formulated as dense tracking objectives, penalising deviation from target geometric references and, in some cases, incorporating smoothness or regularisation terms to ensure stable control behaviour. Both value-based tabular methods and actor-critic algorithms are employed, with deep continuous-control methods such as DDPG and DSAC enabling near real-time multi-input multi-output regulation in simulation or controlled experimental settings.

Only a one study adopts genuinely model-based RL formulations. Other works instead rely on *model-assisted* approaches, where analytical models or reduced-order simulators are used to guide planning or constrain policy optimisation, while the underlying reinforcement learning remains model-free. As shown in Table 2 and Table II in Appendix B, these approaches enable accurate prediction of process responses under limited data availability and support corrective control strategies that reduce error accumulation across layers.

A second major application area concerns deposition path planning and trajectory optimisation. In these studies, the RL state is typically geometric or image-based, encoding layer grids, local fields of view, contour representations, or remaining target regions. Actions are discrete and involve selecting motion primitives, grid movements, layer heights, or trajectory-generation strategies. Reward functions are shaped to balance geometric fidelity, coverage completeness, thermal uniformity, productivity, and motion smoothness. Policy-gradient methods, particularly PPO with convolutional encoders, are widely adopted for these problems, while alternative formulations based on multi-armed bandits are used to accelerate convergence when the decision space can be expressed as a finite set of trajectory options. Reported outcomes indicate reductions in geometric deviation, shorter trajectories, improved scalability, and enhanced deposition reliability compared with heuristic or brute-force approaches.

Table 2
Summary of reinforcement learning applications in DED, including methods, optimisation objectives, and reported outcomes.

Author (Year)	RL/ Optimisation Method	Objective	Key Outcome
Dharmadhikari et al. [123] (2023)	Q-learning	Process parameter optimisation (power–velocity) for melt pool depth control	Predictions within 50 μm accuracy; robust under scarce-data conditions
Dharmawan et al. [130] (2020)	Model-based RL + GPR	Corrective control of MLMB deposition	Reduced error accumulation; improved layer uniformity; near-net-shape builds
Petrik and Bambach [124] (2024)	RL-based optimisation	Deposition path planning for tubular geometries	Reduced layer count by ~30%; improved axis alignment vs. brute-force approaches
Petrik and Bambach [125] (2023)	PPO + optimisation	Joint optimisation of deposition paths and process parameters	Achieved geometric deviation below 0.5 mm; reduced manual intervention; validated on benchmark parts
Sideris et al. [126] (2024)	PPO + Monte Carlo Tree Search + reduced-order thermal model	Deposition path planning balancing productivity & thermal uniformity	Achieved around 20% reduction in temperature deviations; demonstrated trade-offs between efficiency and stability
Ferreira et al. [127] (2024)	MAB (Thompson sampling)	Enhanced-Pixel path optimisation	Achieved shorter trajectories in fewer iterations; reduced computational time; experimental builds free of discontinuities
Ferreira et al. [128] (2025)	MAB (Advanced-Pixel)	Scalable path optimisation for lattice-type geometries	Faster convergence and shorter build times; improved scalability and deposition reliability
Xue et al. [131] (2025)	Deep RL (DSAC)	Droplet volume and shape control	Achieved droplet volume error below 4%; maintained geometric stability; enabled adaptive solder deposition
Mattera et al. [15] (2023)	DDPG + reduced-order modelling	Parameter optimisation and bead geometry control	Enabled near real-time optimisation with accurate bead tracking; highlighted industrial deployment challenges
Li et al. [129] (2023)	Multiclass reinforced active learning (GCN + DQN)	Classification of droplet pinch-off behaviours	Reduced labelling requirement to about 18% of supervised methods; improved classification accuracy; enabled parameter adjustment

Across the reviewed DED literature, substantial variation is observed in the degree of observability and control scope. Some studies rely on direct in situ sensing and continuous feedback, while others operate on simulated or preprocessed geometric representations without explicit thermal state observation. Similarly, the choice of RL paradigm ranges from stateless bandit formulations to fully observable Markov decision processes with continuous states and actions. As evidenced by the outcomes summarised in Table 2, effective performance is achieved across this spectrum, suggesting that successful RL deployment in DED depends primarily on aligning the learning formulation with sensing availability, control authority, and computational constraints rather than on a single dominant algorithmic choice.

3.3. *Fused deposition modelling*

RL has been increasingly applied to extrusion-based additive manufacturing, particularly in FDM, in order to improve toolpath planning, thermal regulation, adaptive control, and defect mitigation. At the pre-processing stage, RL has been employed to optimise toolpath generation. Patrick et al. [132] integrated RL principles into the Nearest Neighbour algorithm to generate infill strategies that reduced bead breaks and printhead lifts, thereby improving print time and structural consistency. Ge et al. [133] developed Q-Path, a Q-learning-based planner tailored for thin-walled structures, which consistently outperformed ZigZag and Fleury heuristics by minimising turns, lifts, and total print length across geometrically complex models. These studies highlight RL's potential to strengthen the planning phase of FDM by producing adaptive, efficient toolpaths.

In terms of process and thermal control, Piovarči et al. [134] extended RL-driven control to Direct Ink Writing (DIW), developing the first closed-loop framework for extrusion processes with complex rheology. A Proximal Policy Optimisation (PPO) agent was trained entirely in simulation, using heightmap observations to adjust nozzle velocity and path offset, and successfully transferred to hardware with improved deposition accuracy and robustness compared to baseline controllers. Mishra and Jatti [135] explored offline Q-learning for parameter optimisation in FDM, identifying optimal combinations of infill, layer height, speed, and temperature that closely matched experimental mechanical performance, achieving good agreement across tensile, flexural, and impact tests. Similarly, Zavrakli et al. [136] applied RL to temperature control in Big Area Additive Manufacturing (BAAM), combining model-free control with Bayesian optimisation for parameter tuning, which improved tracking accuracy and efficiency. Collectively, these works demonstrate RL's versatility across different extrusion-based processes, from fine-scale thermal regulation in FDM to large-scale BAAM deposition.

RL has also been applied to defect mitigation and adaptive quality assurance. Chung et al. [137] proposed Continual G-Learning, a framework that integrates offline knowledge with online adaptation to correct unforeseen defects during printing. By combining image-based defect detection with parameter adjustment, the method achieved defect-free builds in fewer corrective steps than conventional RL approaches, demonstrating its potential for adaptive quality assurance in extrusion-based AM. Cleeman et al. [138] introduced Conditional RL (ConRL) to mitigate defects caused by unpredictable in-field disturbances such as nozzle misalignment or material inconsistency. The approach enabled rapid, single-step corrections and significantly outperformed PID and model predictive controllers, demonstrating robust adaptability to untrained disturbances. Extending beyond extrusion, Wang et al. [139] modelled layer-wise error compensation in Digital Light Processing (DLP) printing as a Markov decision process. Using a Twin Delayed Deep Deterministic Policy Gradient (TD3) agent with convolutional feature extraction, their method reduced pixel-level projection errors by nearly 69% in simulation, outperforming rule-based and neural baselines. Together, these works demonstrate the promise of RL for adaptive defect detection and correction in polymer AM

processes.

Beyond conventional thermal or geometric regulation, RL has been explored for multi-material and 4D printing applications. Liao et al. [140] trained a PPO agent in simulation to optimise feed rates for multi-material deposition using a mixing nozzle, improving the accuracy of material transitions compared with heuristic baselines. Ji et al. [141] applied Q-learning to control shape memory polymers (SMPs), demonstrating improved precision and adaptability in 4D morphing compared to conventional controllers. In a follow-up study, Ji et al. [142] developed an ALQ-based controller, which further enhanced robustness and maintained accurate morphing under material degradation and variability. Mohammadi et al. [143] demonstrated RL-driven control of variable stiffness robotic joints fabricated with SMP springs and carbon-fibre heating elements, achieving reduced energy consumption and enhanced adaptability compared with PID controllers. Collectively, these studies illustrate how RL can extend extrusion-based AM into smart material and 4D printing domains.

Finally, RL has been introduced for design optimisation and large-scale extrusion. Choi et al. [144] employed a duelling deep Q-network to optimise compliant metamaterial mechanisms, producing TPU/PLA structures with significantly improved compliance, validated experimentally. Alghamdi [145] combined RL with neural networks, genetic algorithms, and topology optimisation to balance multi-objective trade-offs such as strength, cost, and print time, achieving up to 30% defect reduction and 25% savings in material and time in PLA prints. Wang et al. [146] proposed an RL-based pointer-network planner for continuous, smooth path generation in 3D concrete printing of complex hollow components; with Bézier smoothing and a greedy multilayer connector, they reduced path redundancy and turning angles, and validated improvements via simulation, FE analysis, and multi-layer printing experiments. As shown in Fig. 11, the proposed method avoided extrusion interruptions and accumulations typical of heuristic strategies, producing smoother and more uniform paths in both simulation and experiments. Although not limited to FDM, Tseng et al. [147] applied deep Q-learning to optimise gyroid composite structures in stereolithography/DLP, achieving superior strength and shock absorption compared with traditional designs.

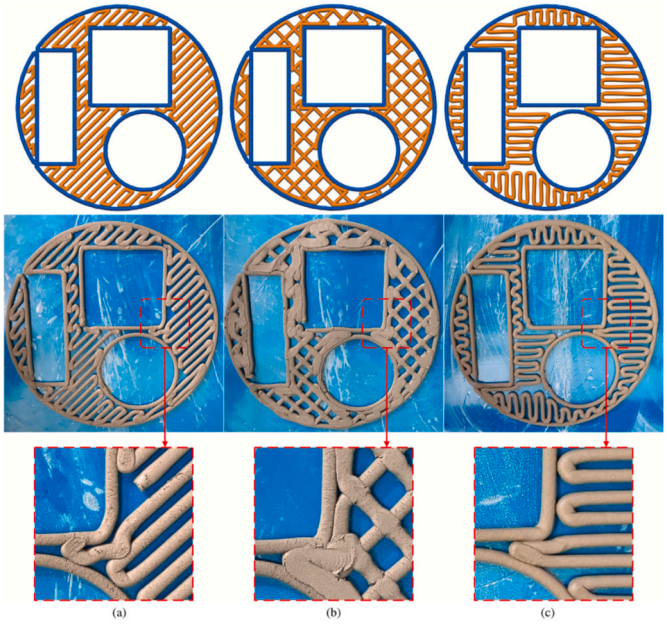

Fig. 11. Comparison of toolpath strategies at 60% fill rate: (a) rectilinear filling showing extrusion interruptions, (b) grid filling showing material accumulation at overlaps, and (c) Q-Path RL planner producing uniform extrusion along a smoothed continuous path. Adapted from Wang et al. [146].

Taken together, these studies reveal the breadth of RL applications in polymer-based AM, spanning FDM, DIW, BAAM, DLP, and SLA. From toolpath optimisation and thermal regulation to defect correction, multi-material deposition, and design exploration, RL has shown its capacity to enhance both efficiency and functionality across extrusion and polymer printing processes. While many frameworks remain at the simulation stage, experimental validations confirm its potential for industrial deployment and adaptive manufacturing. A summary of the reviewed works is provided in Table 3, highlighting the RL methods, objectives, and key outcomes.

While Table 3 summarises RL applications in FDM in terms of methods, optimisation objectives, and reported outcomes, such high-level comparisons do not capture how the underlying RL problems are posed. To address this, Table III in the Appendix C provides a formulation-level comparison of the same studies, explicitly documenting the definition of agent states, action spaces, reward functions, and policy or value-function architectures. This perspective clarifies the diversity of problem abstractions adopted in the FDM literature, ranging from bandit and offline optimisation formulations to fully specified Markov decision processes and optimal control-based RL approaches.

3.3.1. *Comparison analysis*

Tables 3 and Table III in the Appendix C provide a comparative overview of RL applications in FDM and related extrusion-based polymer additive manufacturing processes, highlighting differences in optimisation objectives, RL formulations, and the construction of states, actions, and rewards. Compared with metal-based processes, the reviewed polymer AM studies exhibit a wider range of problem abstractions, reflecting the diversity of extrusion mechanisms, sensing modalities, and control objectives.

A prominent class of applications concerns toolpath planning and pre-processing optimisation. As shown in Table 3 and Table III in the Appendix C, these studies typically formulate the problem using discrete state spaces derived from printhead position, discretised layer grids, or sets of unvisited path points. Actions correspond to selecting the next traversal point, movement direction, or path-generation strategy, while rewards penalise printhead lifts, sharp turns, excessive path length, or incomplete coverage. Value-based tabular methods and bandit-style formulations are common in this category, reflecting the discrete and combinatorial nature of the planning task and enabling efficient optimisation without high-dimensional sensing or deep neural policies.

A second group of studies focuses on process and thermal control, including nozzle velocity regulation, temperature tracking, and extrusion consistency. These works more frequently adopt fully specified Markov decision processes with continuous state and action spaces. States are constructed from geometric measurements, temperature estimates, or image-derived representations such as heightmaps, while actions consist of continuous adjustments to extrusion speed, nozzle velocity, heater power, or material feed rates. Reward functions are typically dense tracking objectives that penalise deviation from target geometry or temperature references and may include additional regularisation terms to promote smooth control behaviour. Deep actor-critic algorithms, particularly PPO, TD3, and related variants, are predominant in this class of problems.

RL formulations aimed at defect mitigation and adaptive quality assurance occupy an intermediate position between planning and control. As detailed in Table III in the Appendix C, states in these studies often encode abstracted defect indicators, such as void fraction, over-extrusion flags, or classified surface quality that were derived from image-based monitoring rather than raw sensor data. Actions involve parameter adjustments applied at discrete decision points, and rewards are sparse or binary, reflecting the achievement of defect-free surface conditions. These formulations prioritise robustness to unmodelled disturbances and rapid corrective action over fine-grained tracking accuracy.

Beyond conventional extrusion control, several studies extend RL to

Table 3
Summary of reinforcement learning applications in FDM, including methods, optimisation objectives, and reported outcomes.

Author (Year)	RL/ Optimisation Method	Objective	Key Outcome
Patrick et al. [132] (2018)	RL-enhanced NN	Toolpath optimisation (infill planning)	Reduced bead breaks and printhead lifts; improved print time and consistency
Ge et al. [133] (2021)	Q-learning (Q-Path)	Toolpath optimisation (reduce turns/lifts, improve efficiency)	Outperformed ZigZag and Fleury; reduced print length, turns, and lifts; improved surface quality
Piovarči et al. [134] (2022)	PPO (closed-loop)	Adaptive control of nozzle velocity & path offset under rheological variability	Improved deposition accuracy and robustness; successful sim-to-real transfer with reduced defects
Mishra and Jatti [135] (2023)	Q-learning	Process parameter optimisation (infill, layer height, speed, temp.)	Achieved error below 11% compared with experiments; improved prediction of mechanical properties
Zavrakli et al. [136] (2023)	Model-free RL + Bayesian optimisation	Temperature control	Improved tracking accuracy and stability vs. PID; better extrusion consistency
Chung et al. [137] (2022)	Continual G-Learning	Defect detection & adaptive parameter adjustment	Achieved defect-free builds; fewer corrective steps than Q-learning and G-Learning baselines
Cleeman et al. [138] (2025)	Conditional RL (ConRL)	Mitigation of defects from unpredictable disturbances (e.g., nozzle misalignment, material inconsistency)	Enabled rapid, single-step corrections; outperformed PID and MPC; robust to untrained disturbances
Wang et al. [139] (2023)	TD3 (Deep RL) + autoencoder feature extraction	Layer-wise error compensation	Reduced pixel-level projection errors by about 69%, outperforming rule-based and UNet baselines
Liao et al. [140] (2023)	PPO + custom deposition model	Feed rate optimisation for material mixing	Improved spatial resolution; reduced mixing artefacts; outperformed handcrafted algorithms
Ji et al. [141] (2022)	Q-learning	Optimal morphing control of SMPs	Improved precision and adaptability in shape morphing vs. conventional controllers
Ji et al. [142] (2022)	ALQ (adaptive LQR-by-Q-learning)	Adaptive morphing control under material variability	Enhanced robustness and accuracy; maintained performance under degradation and variability
Mohammadi et al. [143] (2024)	Deep RL (TD3/ DDPG)	Adaptive stiffness control for robotic joints	Reduced energy consumption by more than 40% and improved adaptability compared with PID controllers
Choi et al. [144] (2025)	Duelling DQN	Design optimisation of compliant metamaterial mechanisms	Generated digitised cell structures with improved compliance; experimentally validated higher deformation efficiency compared with topology optimisation
Alghamdi [145] (2025)	RL + NN + GA + Topology Optimisation	Multi-objective optimisation of print parameters (time, cost, strength, defects)	Achieved 15–25% reductions in print time and material use, a 12% increase in yield strength, and a 30% reduction in defect rates compared to conventional methods
Wang et al. [146] (2025)	RL-based pointer network + Bézier smoothing	Continuous path planning (minimise length & turning) + multilayer connection	Produced continuous, smooth paths; reduced redundancy and turning; improved printing accuracy compared with conventional infill
Tseng et al. [147] (2025)	Deep Q-learning	Structural optimisation of gyroid composite structures	Achieved superior strength and shock absorption compared with traditional designs

multi-material, 4D printing, and design optimisation tasks. In these cases, the RL state frequently represents design configurations or future target patterns rather than instantaneous process measurements, and rewards are often terminal or simulation-based, computed from mechanical, morphing, or functional performance metrics. Both tabular and deep value-based methods are employed, with the choice of formulation reflecting whether the design space is discrete and finite or high-dimensional and continuous.

Overall, the formulation-level comparison in Table 3 reveals substantial variation in the degree of observability, control authority, and learning complexity across polymer AM studies. Effective performance is reported for both low-dimensional, discrete formulations using tabular Q-learning and high-dimensional, continuous formulations using deep actor-critic methods. This diversity indicates that, in extrusion-based polymer AM, the suitability of RL is strongly dependent on how the manufacturing task is abstracted into states, actions, and rewards, rather than on the choice of a single dominant algorithmic framework.

4. Challenges and future perspectives

4.1. Summary of key findings

This review has examined recent applications of RL in additive manufacturing, with an emphasis on recent advances and limitations in feedback control, encompassing powder bed fusion (PBF), direct energy deposition (DED), and extrusion-based processes such as fused deposition modelling (FDM). As a result, it can be summarised that:

- In PBF, where thermal dynamics are fast, spatially localised, and strongly coupled to defect formation, RL is mainly used for melt pool and temperature regulation, with results showing substantial

reductions in tracking error and improved stability compared with classical controllers.

- In contrast, DED results emphasise geometric consistency, deposition path planning, and corrective feedback over longer spatial and temporal horizons, reflecting the greater accessibility of in situ sensing and the slower process dynamics.
- For FDM and other polymer-based extrusion processes, RL has been applied to a wider range of objectives, from toolpath efficiency and defect correction to material mixing and shape morphing, highlighting the flexibility of RL formulations in environments with comparatively lower thermal complexity but higher variability in material behaviour and geometry.

A general observation concerns the relationship between formulation complexity and achieved performance. Across all three process families, strong results are reported for both high-dimensional deep RL formulations and low-dimensional, discretised approaches using tabular learning or bandit strategies. In PBF and DED, several studies demonstrate that physics-informed environments or analytical models can compensate for limited sensing, enabling simple Q-learning formulations to achieve competitive accuracy and robustness. Similarly, in FDM, offline and bandit-based RL approaches achieve meaningful improvements in toolpath quality and parameter selection without requiring deep perception or continuous control. These results collectively suggest that performance gains are driven less by algorithmic sophistication than by careful abstraction of the manufacturing task into appropriate states, actions, and rewards that reflect process physics, sensing constraints, and decision timescales.

Finally, the reported outcomes across all three sections indicate that RL is most mature in structured, well-scoped decision problems, such as single-objective tracking, local corrective control, or constrained path planning, while broader challenges remain for large-scale, fully autonomous deployment. Although many studies demonstrate impressive improvements in simulation or controlled experimental settings, issues such as stability guarantees, sample efficiency, interpretability, and safe exploration recur across processes. Recent results, particularly those incorporating physics-informed rewards, stability-inspired constraints, or sim-to-real transfer, suggest promising directions for addressing these limitations. Taken together, the results reviewed in this paper characterise RL not as a replacement for physics-based modelling or classical control, but as a complementary framework capable of embedding domain knowledge into adaptive decision-making loops across a wide range of additive manufacturing processes. To consolidate the findings across PBF, DED, and FDM, Table 4 summarises representative RL formulations reported in the literature, mapping common additive manufacturing objectives to typical choices of state representation, action space, reward structure, and learning algorithm. The inclusion of model-based formulations is limited to geometry and layer height control, where explicit or learned process models have occasionally been used to support prediction or corrective planning. In most other objectives, reported performance gains are achieved using model-free or model-assisted formulations, with physics knowledge incorporated indirectly through reward shaping or constraints.

4.2. Implementation considerations

With respect to the neural network architectures most commonly adopted in RL-based control, it is important to emphasise that the final control policy is almost invariably implemented as a feedforward neural network. This choice is dictated by the nature of the control action, which is always a discrete or continuous variable. Accordingly, the policy output requires either a SoftMax activation function in the case of discrete actions, or a bounded activation function (e.g. hyperbolic tangent) for continuous action spaces, irrespective of the type of data provided as input to the network.

In classical control theory, image data are never processed directly within the controller. Vision-based PID or MPC strategies rely on upstream computer vision algorithms to extract physically meaningful scalar features, such as layer height, layer width, or melt pool area, which are then used as inputs to the controller. Following the same

Table 4
Representative RL formulations adopted for different AM objectives, based on reported practices across PBF, DED, and FDM studies.

Target Objective	AM Process	Typical RL Formulation	Representative State Definition	Representative Action Space	Representative Reward Structure	Commonly Used RL Methods
Thermal / melt pool stabilisation	PBF, DED	Model-free MDP (continuous)	Melt pool depth/width; temperature field; average layer temperature	Continuous laser power and/or scan velocity updates	Dense tracking reward penalising deviation from target geometry/ temperature, with variance regularisation	PPO, AWAC, DDPG
Multi-parameter thermal control (MIMO)	PBF	Model-free MDP (continuous)	Current and historical melt pool geometry; previous control actions	Joint control of laser power and scan velocity	Weighted tracking error for depth and width, penalising fluctuations	PPO (actor-critic)
Defect suppression (porosity, lack-of-fusion)	PBF, DED	Model-free or model-assisted MDP (discrete)	Discretised process parameters; defect indicators from models or sensors	Incremental parameter adjustments within bounds	Reward proportional to predicted relative density or penalty on defect indicators	Tabular Q-learning, PPO
Geometry / layer height control	DED, FDM	Model-based or model-free MDP (continuous)	Bead width/height; print height from sensors or reduced-order models	Continuous updates to torch speed, wire feed rate, nozzle velocity	Dense geometric tracking reward with smoothness regularisation	DDPG, PPO, DSAC
Toolpath planning and traversal	PBF, DED, FDM	Discrete MDP or bandit	Discretised grid, graph, or visited-node representation	Selection of next node, direction, or path heuristic	Shaped reward penalising turns, lifts, path length, or thermal accumulation	DQN, tabular Q-learning, MAB
Distortion or stress minimisation	PBF, DED	Discrete MDP	Geometric position; proxy thermal accumulation	Discrete move strategies or scan directions	Reward penalising thermal accumulation or sharp turns	DQN
Defect detection & corrective control	FDM, DIW	Model-free MDP (hybrid)	Image-derived defect indicators (voids, over-extrusion flags)	Discrete or continuous parameter corrections	Sparse or dense reward favouring defect-free conditions	Continual G-Learning, TD3
Multi-material / mixing control	FDM, DIW	Model-free MDP (continuous)	Look-ahead target pattern; colour or material composition state	Continuous feed-rate adjustments	Dense reward penalising colour or material mismatch	PPO
Shape morphing / 4D printing	FDM, polymer AM	Model-free MDP	Shape angle; temperature or stimulus state	Continuous temperature or stimulus change	Asymmetric quadratic tracking reward	Q-learning, ALQ

control-theoretic rationale, RL applied to feedback control problems typically operates on scalar state variables rather than raw images. Direct use of CNNs within the feedback loop to generate control actions from images remains rare in the literature and is generally avoided in industrial contexts due to interpretability, robustness, and real-time constraints. As a result, by extension, although RL frameworks can naturally accommodate CNN architectures when spatial data are used as inputs, a scalar state formulation is preferred from a control perspective. This approach ensures that the policy operates on well-defined physical quantities that can be directly compared against reference values, thereby preserving consistency with traditional control methodologies.

Regarding network size, no strict design rules exist. As discussed above, the formulation of the state, action, and reward, as well as the design of the environment (which corresponds to the plant model) are the most critical elements and have the same role as the plant and cost function weights in classical control theory. Therefore, shallow feed-forward networks with one or two hidden layers are generally sufficient for control applications of this type. Rectified Linear Unit (ReLU) activation functions are commonly employed in the hidden layers, while bounded activation functions are systematically adopted at the output layer to enforce admissible action ranges. These design choices are closely linked to operational stability considerations, which are discussed in the following section.

Finally, hardware selection plays a crucial role in enabling real-time deployment of RL-based controllers. The choice depends primarily on the nature of the input data and the computational requirements of the inference phase. Edge devices equipped with GPUs, such as NVIDIA Jetson platforms (e.g. Nano or Orin Nano), are particularly well-suited for RL applications, as they enable fast inference while supporting direct integration with cameras and industrial communication protocols (e.g. TCP/IP or digital I/O). While CPU-based devices such as Raspberry Pi can be used for simple control problems or relaxed timing constraints, GPU-enabled platforms are generally preferred, especially when image-based inputs or more complex policies are considered. This computational advantage represents one of the key reasons why RL can achieve faster online execution than optimisation-based controllers such as MPC, as discussed in Appendix D with a simulation case study.

4.3. Current limitations of reinforcement learning for feedback control in AM

4.3.1. Implementation limitations and possible solutions

Despite the suitability of RL for sequential decision-making and feedback control, its adoption in AM remains limited. Most reported studies focus on toolpath optimisation, process parameter tuning, or off-line optimisation, with relatively few works explicitly applying RL as a closed-loop feedback controller. While RL offers potential advantages over classical control methods, such as the ability to handle nonlinear, multivariable dynamics and to exploit high-dimensional sensor data, many existing demonstrations remain confined to simulation-based environments or tightly controlled experimental settings.

It should be emphasised that, although edge devices and single-board computers such as those previously mentioned are generally preferable for RL-based control, their practical adoption ultimately depends on compatibility with the specific machine architecture. This consideration exposes a broader and more fundamental issue, namely the lack of truly open architectures in current Industry 4.0 implementations. At present, industrial machines often rely on proprietary communication protocols, which vary significantly across manufacturers and applications. As a result, not all edge devices can natively interface with industrial equipment using standardised protocols.

4.3.2. Stability and safety considerations in RL-based feedback control for additive manufacturing

In classical control theory, the introduction of a feedback controller alters the closed-loop dynamics of the system. Also, for simple linear time-invariant (LTI) plant, like

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{35}$$

the application of a proportional or PID controller results in a modified closed-loop system matrix, thereby changing pole locations, time constants, and stability margins. For this reason, stability analysis is a fundamental requirement in any conventional control design. However, in RL-based feedback control approaches for AM, the interpretation of the control action is strongly tied to the architectural assumptions adopted in existing studies. Making these assumptions explicit is crucial to determine when stability can be reasonably assumed without dedicated Lyapunov analysis, and to clarify the transition from classical dynamic stability to operational stability considerations. Accordingly, based on the reviewed literature, we summarise below the main findings that may guide future research directions.

Most existing studies implicitly consider a hierarchical control architecture in which RL acts as a supervisory layer, generating process parameter reference values for embedded low-level controllers (e.g. PID loops). The low-level controllers are assumed to be stable and significantly faster than the supervisory layer. Under this assumption, the aggregated closed-loop process dynamics can be described by a reduced-order discrete-time model of the form

$$x_{t+1} = f(x_t) + g u_t, \tag{36}$$

where:

- $x_t \in \mathbb{R}^n$ represents the system state,
- $u_t \in \mathbb{R}^m$ is the control action (the reference signal) generated by the RL policy π_θ,
- $f(\cdot)$ and g represent the closed-loop dynamics resulting from the embedded control layers.

A crucial implicit assumption underlying the majority of RL applications in AM is that variations in the process parameter reference u_t primarily affect the steady-state operating point of the system, while the transient dynamics remain approximately invariant. Formally, this corresponds to assuming that the Jacobian of the system with respect to the state,

$$J(x) = \frac{\partial f(x)}{\partial x}, \tag{37}$$

is independent of the reference signal u_t within the admissible operating envelope. Under this assumption, for any constant control action (process parameter reference) $u_t \equiv \bar{u}$, the system admits an equilibrium $x^\star(\bar{u})$ satisfying

$$x^\star(\bar{r}) = f(x^\star(\bar{u})) + g\bar{r}, \tag{38}$$

and the local stability properties around each equilibrium are governed by the same linearised dynamics. Consequently, stability is preserved independently of the selected set-point, and classical Lyapunov stability analysis is not required. It should be noted that this assumption is rarely stated explicitly and is not supported by formal theoretical guarantees in the existing literature. Nevertheless, it represents a reasonable approximation in many AM scenarios, where material properties, feedstock geometry, and process modality are fixed, and the dominant physical dynamics (e.g. thermal diffusion and cooling rates) are primarily material-dependent. In this case, within restricted operating ranges, moderate variations in process parameters are therefore assumed to shift the steady-state operating point without significantly altering the system dynamics.

When the above assumption holds, the control problem shifts from guaranteeing closed-loop stability to ensuring operational stability. In this setting, instability does not manifest as divergent state trajectories,

but rather as undesirable process behaviours arising from incompatible or excessively variable reference sequences. Consequently, stability-related issues in AM are more accurately framed in terms of operational stability rather than classical Lyapunov stability. To mitigate these issues, several practical mechanisms can be adopted:

- Bounding of control actions, ensuring that the learning agent operates within a restricted and physically meaningful parameter envelope. In some works, the output activation function of the control policy is bounded via tanh or sigmoid functions.
- Penalisation of rapid variations in control actions, typically implemented through rate-of-change terms in the reward function. This can be formalised by introducing a regularisation term on the rate of change of the control action,

$$\Delta u_t = u_t - u_{t-1}, \tag{39}$$

which can be incorporated into the RL reward function as

$$r_t = \mathrm{r}_t(x_t, u_t) - \lambda \parallel \Delta u_t \parallel_p, \tag{40}$$

where $p \in \{1, 2\}$ and $\lambda > 0$ is a weighting coefficient. In particular, L1 regularisation promotes sparse and conservative updates of the reference signal, which is especially relevant in multi-input settings.

- Hybrid surrogate models, integrating regression models for process prediction with classification or anomaly detection components aimed at identifying infeasible or unsafe operating regions. Specifically, beyond approximating the steady-state response of the process, the surrogate model should include a mechanism to evaluate whether a given control action is expected to result in a stable or unstable operational condition at steady state.

The assumptions discussed above become less evident in nonlinear or MIMO configurations, where the system dynamics may depend explicitly on the control action. In such cases, the system must be described by a general nonlinear, data-driven model

$$x_{t+1} = f(x_t, u_t), \tag{41}$$

where different control actions may induce qualitatively different dynamic behaviours. In this scenario, stability can no longer be assumed a priori, and Lyapunov-based considerations become relevant. For discrete-time systems, a Lyapunov function $V(x)$ satisfies

$$V(x) \geq 0, V(0) = 0, \tag{42}$$

and

$$\Delta V_t = V(x_{t+1}) - V(x_t) < 0, \tag{43}$$

ensuring convergence towards a stable operating condition. In RL settings, where the system dynamics are unknown, Lyapunov-based constraints can be incorporated implicitly by augmenting the reward function with a stability-related penalty term. A common practical choice is

$$V(x_t) = \parallel x_t \parallel^2, \tag{44}$$

leading to

$$\Delta V_t = \parallel x_{t+1} \parallel^2 - \parallel x_t \parallel^2. \tag{45}$$

Negative values of ΔV_t correspond to dissipative behaviour and can be rewarded, whereas positive values are penalised. This approach allows the learning agent to discourage destabilising actions without requiring an explicit analytical model of the system dynamics. As a result, another penalisation term can be added

$$r_t = \mathrm{r}_t(x_t, u_t) - \lambda \parallel \Delta u_t \parallel_p - \Delta V_t \tag{46}$$

To summarise, stability considerations in RL–based feedback control for AM depend critically on the underlying modelling assumptions. When process parameters are assumed to affect only the steady-state operating point, stability is guaranteed by construction, and the focus should be placed on operational constraints. Conversely, when control actions influence the system dynamics, Lyapunov-inspired mechanisms provide a principled means of enhancing safety and robustness in data-driven control frameworks. By contrast, in other RL applications such as process monitoring or tool-path optimisation, stability issues are not of primary concern, since RL is not used as a controller but as an optimisation tool for tasks that do not involve dynamic closed-loop behaviour.

4.3.3. Scaling from sim-to-real

Another barrier lies in the fact that training RL agents often requires a very large number of interactions, which is impractical in AM due to the high material and time costs involved. In particular, during the early stages of training, the agent's actions are essentially random, which increases the risk of equipment damage or process failure. Indeed, RL as a field has experienced periods of rapid growth, especially following its success in solving complex yet well-defined problems such as strategic games, followed by a "winter" when its direct application to physical systems proved considerably more challenging. Unlike games, where uncertainty can be explicitly modelled and controlled, AM processes exhibit highly complex, partially stochastic dynamics that are often too intricate to be fully modelled with high fidelity. This complexity makes accurate state representation and prediction difficult, hindering the direct transfer of RL algorithms from simulation to physical AM systems. As a result, significant challenges remain in ensuring stability, sample efficiency, and robustness when deploying RL in real-world AM contexts. One possible approach to mitigating these challenges is the use of simulation-to-reality (sim-to-real) transfer and reduced-order models (ROMs), which can help bridge the gap between simulated and physical systems. However, performance degradation still occurs due to inevitable mismatches between the simulated environment and the real process. Another promising strategy is to pre-train the control policy in a simplified state-action setting or to initialise it using human-informed methods, so that the network does not start from a completely random policy but rather from an informed initial guess based on expert knowledge. This pre-training stage could take the form of supervised learning, after which the policy could be fine-tuned through interaction with a highly realistic ROM of the environment, incorporating both uncertainty modelling and stochastic dynamics. Such a staged approach could improve sample efficiency, reduce equipment risk, and accelerate convergence to effective control policies. Nevertheless, these strategies remain largely unexplored in the current literature, and further research in this direction could yield significant advancements in applying RL to AM feedback control.

4.4. Future research directions

4.4.1. Going beyond simple set-point control

A further perspective is to move beyond simple set-point regulation. Much of the current RL-for-feedback-control literature in AM concentrates on substituting, or modestly improving upon, single-loop set-point controllers to cope with nonlinearity and multivariable coupling. However, RL can be positioned at a higher supervisory layer, closer to MPC, to make temporally coherent, constraint-aware decisions that consider broader system objectives. As demonstrated in a couple of exciting works, RL would not merely deliver timely control actions; it could also integrate information from other software modules within the cyber-physical stack, including process monitoring, soft sensors, anomaly-detection modules, quality predictors, and scheduling/dispatch systems. These signals can enter the RL formulation as part of the state, as auxiliary constraints, or through multi-objective reward functions (e.g., set-point accuracy, energy use, quality), enabling decisions that are jointly optimal for process quality and production

performance.

Despite this high potential, the literature offers few demonstrations that genuinely transcend set-point control in industrial AM. Comprehensive case studies integrating RL with MPC, plant-wide monitoring, and quality/scheduling modules under explicit safety and constraint handling remain scarce. Developing such end-to-end, constraint-aware, system-level integrations is therefore a promising direction for future research and industrial uptake.

4.4.2. Open architecture for easier implementation

One potential solution to the interoperability issue between devices in AM may be the adoption of OPC UA, which provides a standardised framework for industrial data exchange over TCP/IP and can, in principle, be enabled on the edge devices discussed above. However, in practice, not all AM systems are currently capable of receiving or exposing process data through OPC UA interfaces. While robotic motion platforms are generally well supported, as most major robot manufacturers provide relatively standardised communication interfaces or ROS-compatible drivers, integration becomes considerably more complex when dealing with process equipment such as power sources for welding or laser systems. In some AM technologies, such as FDM, it may be possible to replace proprietary machine software with open-source solutions to enable advanced control strategies. In contrast, for welding-based or laser-based systems, this level of access is often not available, and interaction with the process must be mediated through the robot controller or other intermediate layers. This indirect access significantly increases system complexity and limits the feasibility of deploying RL-based controllers in real industrial environments. These integration challenges partly explain why, despite the growing body of academic research, there are currently very few industrial implementations of online reinforcement learning-based control. The main obstacles are not only algorithmic but also practical, encompassing system integration, communication constraints, and stability considerations. The latter aspect represents a critical challenge and is discussed in more detail in the following section.

4.4.3. Real case studies and integration with digital twin architecture

Future research should increasingly focus on real industrial case studies in which RL controllers are tightly integrated within digital twin (DT) architectures, moving beyond proof-of-concept simulations. While model-based control strategies such as MPC remain effective for well-modelled and stationary processes, their reliance on accurate and continuously updated system models limits their scalability in complex manufacturing environments.

In contrast, RL is particularly well-suited to operate within a DT framework because it does not require an explicit analytical model of the process dynamics. Instead, the DT can serve as a high-fidelity, risk-free environment in which the model-free RL agent can be trained, validated, and stress-tested under a wide range of operating conditions, disturbances, and process drifts that would be impractical or unsafe to explore on the physical system. This enables extensive policy optimisation without interrupting production or compromising quality.

Moreover, RL can naturally exploit the bidirectional information flow enabled by DTS. On the one hand, the DT provides synthetic data, scenario generation, and accelerated experience for policy learning. On the other hand, the RL agent can enhance the DT itself by identifying control-relevant patterns, adapting to previously unseen regimes, and informing model updates through observed discrepancies between simulated and real behaviour. This mutual reinforcement is considerably harder to achieve with MPC-based approaches, which typically depend on fixed model structures and require explicit re-identification procedures when system dynamics change.

In the context of AM, where processes are highly nonlinear, layer-dependent, and subject to cumulative thermal and material effects, the combination of RL and DTs offers a real promising pathway towards adaptive, self-improving control strategies. Future developments should therefore prioritise hybrid DT-RL frameworks, systematic validation on real systems, and the definition of performance and safety guarantees that facilitate industrial adoption. Indeed, at the current stage, research efforts in this direction remain limited and are largely confined to simulation-based studies, with little evidence of significant industrial impact or real-world deployment. Most existing approaches address relatively simple control objectives, such as set-point tracking, which restricts their practical relevance. However, as discussed, the integration of RL within DT architectures coupled with real-time monitoring systems has the potential to substantially extend these capabilities, enabling more adaptive and context-aware control strategies and representing a promising direction for future developments.

5. Conclusions

Reinforcement learning (RL) has recently been explored as advanced learning techniques to improve robotic additive manufacturing (AM) processes with adaptive, data-driven feedback control. This review surveys both traditional and RL-based applications across powder bed fusion (PBF), direct energy deposition (DED), and extrusion-based methods (e.g., fused deposition modelling, FDM), based on a structured search of the literature from 2018 to 2025. We find that most studies employ RL for toolpath planning and process-parameter tuning, whereas comparatively few address closed-loop feedback control; those that do are predominantly simulation-based, with a smaller but growing body of experimental validations. Three structural barriers explain this gap to industrial deployment. First, stability, safety, and certification are not yet addressed with the same rigour as in classical control theory; black-box policies and on-line adaptation complicate verification and potentially qualification of the parts. Second, training at scale demands interactions that are prohibitively costly on real machines, and transferring policies from simulation may suffer from model mismatch. Third, integration on production hardware is constrained by closed, vendor-specific ecosystems, millisecond-scale real-time deadlines, limited write access, and the realities of edge inference under power and thermal budgets.

Looking ahead several directions appear particularly promising to solve those problems. First, hybrid control architectures that combine RL with classical controllers (e.g., PID/LQR/MPC) can balance adaptability with stability: RL operates at the supervisory level for tasks such as set-point scheduling, constraint tuning, or mode switching, while certified feedback loops manage low-level actuation, thereby mitigating stability concerns and easing certification. Second, coupling RL with digital twins and reduced-order models (ROMs) offers a route to sample-efficient and safe development: high-fidelity twins enable off-line pre-training, scenario stress-testing, and policy verification prior to deployment, whereas ROMs provide fast surrogates for on-line planning and fine-tuning. Third, beyond process control, embedding RL within the broader cyber-physical AM ecosystem, including design optimisation, anomaly detection and quality prediction, production scheduling, and predictive maintenance could enable end-to-end autonomy. Such system-level integration would allow multi-objective optimisation (e.g., quality, throughput, energy, and cost) under explicit safety and regulatory constraints, moving towards truly autonomous, adaptive, and resilient AM systems.

CRediT authorship contribution statement

Giulio Mattera: Writing – review & editing, Writing – original draft, Investigation, Formal analysis, Data curation, Conceptualization. **Elena Manoli:** Writing – review & editing, Writing – original draft, Investigation, Formal analysis, Conceptualization. **Ethan Canzini:** Writing – review & editing, Writing – original draft, Investigation, Formal analysis, Conceptualization. **Luigi Nele:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The corresponding author Giulio Mattera declares to accept responsibility for all statements listed below:

- All authors have given their contribution in the drafting of this original paper;
- All authors concur with the submission;
- I have taken the consent from all co-Authors before publishing the present article in this journal
- The manuscript has not been submitted to another journal and will not be published elsewhere within one year.
- All authors have no financial/commercial conflict of interest of any kind to disclose.

Appendix A. RL Application in PBF

Table I
Formulation-level comparison of RL approaches applied PBF, detailing the definition of states, action spaces, reward functions, and policy or value-function architectures.

Author	RL Formulation	State Representation	Action Space	Reward Definition	Architecture
Wasmer et al. [110] (2018)	Model-free MDP	Acoustic-emission wavelet time-frequency spectrograms extracted from 160 ms signal windows; however, the paper does not specify how these spectrograms are mapped to a discrete or continuous agent state	Actions are described abstractly as state-to-state transitions within the MDP framework, but the concrete action set is not explicitly defined	The paper formulates an RL objective based on expected discounted reward, but no explicit reward function or numerical reward definition is provided	No neural network or explicit function approximator is described; learning is based on Q-learning with ε-greedy exploration and tabu-search-based exploration control
Knaak et al. [118] (2021)	Model-based MDP	State vector defined as $s_t = (P_t, v_t, Sa_{mean,t}, \delta_t)$ where laser power and scan velocity are directly measured, and surface roughness and defect percentage are estimated from high dynamic range (HDR) optical images using a CNN-based vision pipeline trained on image patches	Discrete action set consisting of incremental changes to laser power and scan velocity (increase, decrease, or no change), forming a finite action space of eight possible control actions applied layer-wise	A reward function that combines terminal rewards and continuous penalties, rewards low surface roughness and low defect percentage, and strongly penalise defect levels exceeding 10%	Model-based RL with a Random Forest regression model used as a learned dynamics function; CNN used for state estimation; MPC used for action selection; Q-learning used as a baseline comparator
Malik et al. [119] (2024)	RL-assisted parameter tuning within a digital twin framework	Input to the RL agent is a vectorised representation of real-time sensor data, pre-processed via normalisation, dimensionality reduction, and feature selection; the paper does not explicitly define an RL state vector or observation model	Actions correspond to incremental adjustments (increase/decrease) of LPBF process parameters (e.g., laser power, scan speed, layer thickness) to drive predicted lack-of-fusion metrics into a controllable range; action bounds and discretisation are not explicitly specified	Reward signal is implicit and associated with maintaining the predicted lack-of-fusion indicator within an acceptable threshold; no explicit mathematical reward function or discounting formulation is provided	PPO with a multilayer perceptron with two fully connected hidden layers of 64 units each. RNN is used solely for defect prediction (state estimation), not as part of the RL policy
Ogoke and Barati Farimani [120] (2021)	Model-free MDP	Multi-view local temperature observations, represented by nine 2D heat maps capturing $x-y, x-z$, and $y-z$ cross-sections around the laser for the current and two previous timesteps	Laser process parameter updates, including laser velocity or laser power, rescaled to bounded ranges	The normalised absolute error between the target melt pool depth and the current melt pool depth, with an additional regularisation term penalising large melt pool depth variation within an episode	Deep neural network policy optimised using PPO; two-layer fully connected network with 64 neurons per layer and tanh activation functions
Grais et al. [121] (2023)	Model-free MDP for error compensation	Continuous state composed of current melt pool depth and width, the depth and width of the three previous melt pools, the corresponding previous control actions, and the current beam location along the scan path	Continuous action space consisting of laser beam power and scan velocity, bounded within predefined operating ranges and applied at each time step	Shaped reward combining normalised tracking error for melt pool depth and width relative to target values, with additional regularisation terms penalising large depth and width fluctuations within an episode	Actor-critic neural network architecture trained with PPO; fully connected policy and value networks with two hidden layers of 64 neurons each and tanh activation functions
Vagenas and Panoutsos [122] (2023)	Model-free MDP (stability-focused analysis)	State defined as a continuous temperature field around the melt pool, represented as a $(9, 10, 10)$-dimensional tensor; state definition is inherited directly from the referenced prior work and not modified	Continuous action corresponding to laser power $0W \leq P \leq 500W$, applied at each time step along a predetermined scanning path; action definition is adopted from prior work	Per-step reward based on normalised error between actual and target melt depth, with an optional heuristic penalty on melt depth variance; reward formulations are reused for comparative stability analysis and not proposed as novel	PPO used as a baseline RL algorithm for the case study; policy/value function parameterisation is not the focus of the paper and architectural details are not fully specified
Venugopal and Anand [111] (2023)	Multi-armed bandit (UCB-based)	No explicit RL state defined; the method operates directly on SIMP topology optimisation variables, where historical element-wise	Implicit element-wise density update actions embedded within the SIMP sensitivity-based update rule; actions correspond to increasing or	Objective defined by compliance minimisation or thermal resistance minimisation with additive manufacturing constraints;	UCB multi-armed bandit strategy integrated into SIMP-based topology optimisation; no learned policy, no value function, no neural network

(continued on next page)

Table I (*continued*)

Author	RL Formulation	State Representation	Action Space	Reward Definition	Architecture
		density update trends implicitly guide exploration	decreasing material density during each topology optimisation iteration	diversity encouraged indirectly via an UCB exploration term rather than an explicit reward signal	
Huang et al. [112] (2024)	Model-free MDP for graph-based path planning	Constructed local graph state on an on-the-fly Local Search Graph (LSG), represented as a 3D tensor $S_t = [A_t, A_{t-1}, A_{t-2}]$of adjacency matrices encoding node connectivity, normalised edge lengths, visited-edge masking, and short-term memory	Discrete action selecting the next node to visit within the current LSG; execution restricted to one-ring neighbours of the current node	Explicit, application-dependent step reward reflecting manufacturing objectives (e.g., collision avoidance, turning-angle penalties, temperature accumulation), with a non-uniform Gaussian-shaped discount over lookahead depth	Deep Q-Network with graph-structured convolutional layers (Edge-to-Edge and Edge-to-Node) followed by fully connected layers; target network and experience replay used during training
Qin et al. [113] (2024)	Model-free episodic MDP solved via value-based DQN	Low-dimensional geometric state consisting only of the current laser position(x_t, y_t); thermal fields are computed internally by a physics-based simulation for environment updates and reward evaluation but are not part of the RL observation	Discrete action selecting one of three predefined next-move strategies: minimum thermal accumulation, smoothest continuation (largest turning angle), or second-smoothest continuation; actions are constrained by geometric boundaries and collision rules	Shaped per-step reward based on a geometric proxy for thermal accumulation, penalising closely spaced successive turns below 90°; additional penalties applied for excessive collisions and isolated points; episodic return accumulated over the full toolpath	Deep Q-Network with fully connected multilayer perceptrons for main and target Q-functions (two hidden layers), three output actions, experience replay, and target network updates
Vagenas et al. [114] (2024)	Model-free episodic MDP	Continuous low-dimensional state comprising the observed average layer temperature, previously applied laser power, and current build height (layer index); state variables are obtained from a control-oriented physics-based SLM simulation	Continuous control action corresponding to the laser power applied at each layer, bounded within predefined ranges (e.g., 250-300 W or 250-350 W depending on the case study)	Dense per-layer reward defined as a normalised tracking objective based on deviation from a target average layer temperature; for the AWAC variant, an additional Lyapunov-inspired stability cost based on meltpool area is introduced during policy updates without modifying the reward function	Actor-critic deep reinforcement learning (Soft Actor-Critic and Adaptive Weighted Actor-Critic) implemented with fully connected neural networks for policy and value functions; no perception networks are used
Park et al. [115] (2025)	Model-free episodic MDP	Discrete low-dimensional state $s_t = [k_1, k_2]$with eight total states, where $k_1$encodes upcoming path geometry (straight vs. acute turnaround) based on inter-point distance, and $k_2$encodes discretised melt pool measurement error relative to a reference, obtained via heuristic binning of features extracted from NIR images; raw images are not part of the RL state	Discrete action set consisting of laser power offsets relative to a nominal open-loop power, applied at each scan point and bounded within predefined limits	Dense per-step shaped reward defined as an asymmetric piecewise linear function of melt pool measurement error, penalising overheating more strongly than under-melting, with the objective of reducing tracking error and variability	Tabular Q-learning with an explicit Q-table over a finite discrete state-action space; no neural network policy or value-function approximation is used
Faizan Mohamed et al. [116] (2025)	Model-free episodic Markov Decision Process MDP	Discrete state space where each state corresponds to a unique combination of three discretised process parameters: laser power (P), scan speed (v), and hatch spacing (h), defined on a fixed parameter grid. No physical sensing: state transitions and porosity outcomes are evaluated using a physics-informed analytical model based on the Eagar–Tsai thermal formulation	Discrete action space consisting of all feasible combinations of incrementing, decrementing, or keeping constant each of the three parameters, excluding the null action; total of 26 actions per state	Dense per-step reward computed from a physics-informed porosity prediction model: reward combines normalised dimensionless melt pool indicators $((\pi_1, \pi_2))$ associated with lack-of-fusion and keyhole porosity, with additional penalties on excessive volumetric energy density; higher reward corresponds to higher predicted relative density	Tabular Q-learning with an explicit Q-table over the finite state–action space; no function approximation or neural network policy/value representation is used. The analytical porosity model serves as the environment, not the RL architecture
Yin et al. [117] (2025)	Model-free episodic MDP	Discrete state defined by the current process parameter pair (P_t, v_t), where laser power and scan velocity are discretised over a finite grid; no online sensing is used, and molten pool depth is predicted offline using an analytical Eagar-Tsai-based thermal model	Discrete action set consisting of incremental adjustments to laser power and scan velocity within predefined bounds	Dense per-step reward based on deviation between predicted molten pool depth and a target depth, with higher reward for smaller absolute error and terminal states defined by convergence to the target region	Tabular Q-learning with an explicit Q-table over the discretised state–action space; no function approximation or neural network is used

Appendix B. RL application in DED

Table II
Formulation-level comparison of RL approaches applied to DED detailing the definition of states, action spaces, reward functions, and policy or value-function architectures.

Author	RL Formulation	State Representation	Action Space	Reward Definition	Architecture
Dharmadhikari et al. [123] (2023)	Model-free MDP	Discrete state defined as the current process parameter pair $s_t = (P_t, v_t)$, where laser power and scan velocity are discretised over a predefined $P - v$ grid. Melt pool depth is not sensed online but evaluated via an analytical Eagar-Tsai-based digital twin, which constitutes the environment.	Discrete action set of eight actions corresponding to simultaneous incremental changes in laser power and scan velocity $(\Delta P \Delta v)$within the discretised grid.	Shaped scalar reward based on deviation between predicted steady-state melt pool depth and a target depth, with higher reward for smaller absolute error and penalties for deviation beyond a tolerance.	Tabular Q-learning with an explicit Q-table over the discrete state-action space; no function approximation or neural network is used.
Dharmawan et al. [130] (2020)	Model-based MDP	Observed state defined as local print height, obtained from 3D point clouds measured using a line laser scanner; the agent's state is computed as the mean print height within a radius δ mm around each discretised print-path waypoint	Continuous control actions consisting of torch speed and wire feed rate, selected at each waypoint and constrained within the process operating window	Explicit reward function penalising deviation from the desired print height and incorporating an exploration term proportional to the prediction uncertainty of the learned dynamics model	Gaussian Process Regression (Kriging) dynamics model used for state transition prediction; no neural network
Petrik and Bambach [124] (2024)	Model-free MDP	Image-based state derived from a 2D parametrisation of the bent tube geometry, encoding inner and outer contours, current layer position, and remaining contour lengths; multiple consecutive fields of view are stacked and processed via a CNN to form the agent state	Discrete action space corresponding to the selection of layer heights $(h_1)$and $(h_2)$for the inner and outer contours at the next layer, with values discretised between 0.5 mm and 3 mm	Explicit multi-term shaped reward penalising non-perpendicular layer orientation, height imbalance within a layer, and frequent height changes, while rewarding balanced reduction of remaining inner and outer contour lengths and consistent layer progression	Actor-critic PPO architecture consisting of a CNN feature extractor followed by separate MLP actor and critic heads; all network layers, activation functions, and training hyperparameters are explicitly specified
Petrik and Bambach [125] (2023)	Model-free MDP	Discretised 2D layer grid; observation provided as a local field-of-view (FOV) window around the agent, represented as a two-channel image encoding (i) current agent position and (ii) remaining target centre-line pixels to be visited/covered; centre line obtained via geometric preprocessing (e.g., discretisation/skeletonisation) and flattened before MLP encoding	Discrete motion actions on the grid; process parameters (e.g., welding speed, wire feed rate) are not RL actions and are determined separately via an optimisation routine	Shaped reward with positive reward for visiting a target centre-line cell, and penalties for revisiting already deposited cells, reaching the boundary, and changing direction; exact penalty magnitudes/weights are not fully specified in a single explicit numeric reward equation in the paper	Deep actor-critic with MLP encoder and separate MLP actor and critic heads (no tabular value function)
Sideris et al. [126] (2024)	Model-free episodic MDP	Image-based discrete state defined on a 2D grid representation of the layer geometry; the state consists of stacked binary images encoding weldable regions, visited cells, and the current agent position	Discrete actions corresponding to grid movements of the deposition head, applied sequentially to construct a continuous path covering the entire layer	Explicit shaped reward: positive reward for reaching target weld regions and completing full layer coverage in a single run, with penalties for laser shut-offs, direction changes, and incomplete coverage	Actor-critic PPO architecture with a CNN feature extractor followed by separate MLP actor and critic heads; CNN and MLP layer sizes, activations, and training hyperparameters are explicitly specified
Ferreira et al. [127] (2024)	Multi-armed bandit	No explicit RL state is defined. The algorithm operates directly on a fixed set of predefined trajectory "options". Historical performance of each option is tracked implicitly through reward statistics rather than a state vector	Discrete action corresponding to selecting one option from a finite set of candidate trajectory-generation heuristics (10 options per iteration) to generate a single trajectory at each iteration	Scalar reward defined from trajectory distance outcomes: options yielding shorter trajectories are treated as rewards, while longer trajectories contribute to regret. Learning objective is to minimise cumulative regret and accelerate convergence towards minimal trajectory length	Thompson sampling-based multi-armed bandit algorithm with Bayesian beta distributions to model option performance. No policy network, value function approximation, or neural network is used
Ferreira et al. [128] (2025)	Multi-armed bandit	No explicit RL state. Decision-making is stateless and based solely on accumulated performance statistics of each arm (historical trajectory distances)	Discrete actions corresponding to selecting one of 10 predefined axis-ordering combinations for trajectory generation at each iteration	Scalar reward derived from trajectory distance produced by the selected AO-HTP combination; objective is minimisation of path length, implemented via negative reward or regret-based formulations; cumulative regret used for evaluation	Multi-armed bandit value estimation using tabular statistics (Q(h)) and selection counts (N (h)); ε-greedy, UCB, and Thompson Sampling policies are evaluated. No policy network, value function approximator, or neural network is used

(*continued on next page*)

Table II (*continued*)

Author	RL Formulation	State Representation	Action Space	Reward Definition	Architecture
Xue et al. [131] (2025)	Model-free continuous-state, continuous-action MDP	Continuous state vector comprising droplet volume error, droplet aspect ratio, and previously applied waveform parameters; droplet features are extracted from in-situ vision using deterministic image processing and PCA, with raw images not used as the RL state	Continuous action defining the period and amplitude of the next driving waveform, bounded within predefined operating ranges	Dense quadratic reward penalising droplet volume tracking error and excessive elongation, prioritising accurate volume control with secondary shape regularisation	Distributional Soft Actor-Critic (DSAC) actor–critic with fully connected neural networks for policy and value distribution approximation; Gaussian policies with entropy regularisation and target networks are employed
Mattera et al. [15] (2023)	Model-free continuous-state, continuous-action MDP	Continuous low-dimensional state comprising measured bead width and height, corresponding references, and previously applied control actions; geometry obtained from a reduced-order data-driven process model	Continuous action vector of process parameter updates (welding speed, wire feed speed, voltage, nozzle-to-workpiece distance), bounded within predefined limits	Dense per-step tracking reward based on deviation between measured and target bead geometry (width and height), using a bounded polynomial reward to ensure smooth gradients near the setpoint	Deep Deterministic Policy Gradient (DDPG) with fully connected actor-critic networks, target networks, and experience replay; reduced-order models and simulators form part of the environment, not the RL architecture
Li et al. [129] (2023)	Model-free MDP	Explicit state characterising the unlabelled image pool, constructed as a three-column matrix comprising (i) graph density computed from GCN-extracted image features to capture data diversity, (ii) classification margin to represent model uncertainty, and (iii) an indicator vector encoding whether images have been labelled	Discrete action defined as selecting one unlabelled image from the pool for human annotation at each query iteration	Composite reward consisting of an extrinsic reward defined as the reduction in cross-entropy loss on a validation set between consecutive iterations, and an intrinsic reward encouraging balanced classification performance across all pinch-off behaviour classes	Deep Q-network (DQN) with duelling architecture, double Q-learning, and prioritised experience replay; GCN used for feature extraction; MLP used as classifier

Appendix C. RL application in FDM

Table III
Summary of reinforcement learning applications in FDM including methods, optimisation objectives, and reported outcomes.

Author	RL Formulation	State Representation	Action Space	Reward Definition	Architecture
Patrick et al. [132] (2018)	Bandit / episodic RL-inspired formulation	State representation: Implicit procedural state corresponding to the current printhead position and unvisited toolpath points; no explicit state vector or observation model is defined.	Discrete selection of the next toolpath starting point or traversal order, based on nearest-neighbour heuristics	Explicit reward defined as the negative number of printhead lifts (bead breaks) associated with a generated toolpath	No neural network or learned function approximator; heuristic-based RL built on a modified recurring nearest-neighbour (RNN) algorithm
Ge et al. [133] (2021)	Model-free MDP	Discrete state defined as the current position of the print head within a 2D discretised layer environment, where each state corresponds to a printing control point	Discrete action set corresponding to transitions from the current control point to neighbouring printable control points in the discretised layer	Reward function composed of negative penalties for print-head turning and lifting actions, where lifting incurs a significantly higher penalty than turning, and the objective is to maximise the total accumulated reward while traversing all states	Tabular Q-learning with ε-greedy exploration; no neural network or function approximation employed
Piovarči et al. [134] (2022)	Model-free MDP	Explicit engineered observation space derived from in-situ vision: raw images captured by a dual-camera optical system are geometrically calibrated, stitched, and converted into a local heightmap cantered at the nozzle; the final state is a 3-channel image comprising the local heightmap, target print geometry, and baseline path, aligned with the print direction	Continuous high-level control actions consisting of printing-head velocity and lateral displacement from the baseline printing path, bounded by hardware constraints	Dense, shaped reward computed in simulation using privileged information, rewarding deposition within the target region, penalising over-deposition and under-deposition, and encouraging uniform height; reward is defined over long horizons via delta rewards	Convolutional neural network policy trained using PPO; CNN processes the image-based state directly and outputs continuous control actions
Mishra and Jatti [135] (2023)	Q-learning-based parameter optimisation (offline, dataset-driven)	Discrete state defined as a tuple of process parameters (infill percentage, layer height, print speed, extrusion temperature), where each state corresponds to one experimentally evaluated parameter combination in a fixed dataset	Discrete actions corresponding to selecting alternative process parameter combinations from a predefined finite set; actions do not represent incremental control inputs applied during printing	Immediate scalar reward equal to the experimentally measured mechanical property (tensile, flexural, or impact strength, depending on the optimisation objective), obtained directly from the dataset	Tabular Q-learning with an explicit Q-table over a finite discrete state–action space; no function approximation, neural network, or environment model is used

(*continued on next page*)

Table III (*continued*)

Author	RL Formulation	State Representation	Action Space	Reward Definition	Architecture
Zavrakli et al. [136] (2023)	Optimal control-based RL (LQT)	System state is not directly observable. In the model-based case, the state is estimated from temperature measurements using a Luenberger observer; in the data-driven case, an augmented state is reconstructed from a finite history of past inputs, outputs, and reference signals	Continuous control inputs corresponding to heater power levels and motor input in a BAAM extruder (multi-input control vector applied at each time step)	Quadratic tracking cost penalising deviation between measured output and reference signal, together with control effort, accumulated over an infinite horizon (reward defined as the negative cost)	Linear state-feedback controller derived from a learned quadratic value function. Value function kernel matrix estimated via least-squares Bellman updates using value iteration; no neural network or nonlinear function approximation is used
Chung et al. [137] (2022)	Model-free MDP	Discrete state defined as the combination of process-parameter levels (flow rate multiplier, printing speed, cooling fan status). State observability is achieved indirectly through online surface images captured by a digital microscope, which are classified as target quality or defect using an offline-trained image-based classifier.	Discrete actions consisting of incrementing or decrementing a single process parameter (e.g., increasing/decreasing flow rate, speed, or toggling cooling fan) at each decision step.	Sparse binary reward: positive reward when the classified surface quality matches the target surface quality; zero otherwise. Reward is determined via majority voting over a sliding window of 21 consecutive surface images to improve robustness against noise.	Tabular G-Learning / Continual G-Learning (model-free RL with prior policy regularisation). No neural network policy or value function. CNN (pre-trained ResNet) is used only for feature extraction in the surface-quality classifier; one-class SVM used for defect detection.
Cleeman et al. [138] (2025)	Model-free continuous-state, continuous-action MDP	Continuous defect state ($s_t = D_t = [VF_t, OE_t]$), where (VF_t) is planar void fraction and (OE_t) is a binary overprinting indicator. Defect state is obtained from in-situ images via CNN-based defect classification followed by deterministic image segmentation and quantification; raw images are not used as RL state	Continuous action corresponding to the commanded filament speed (F_{t+1}), bounded within predefined limits; policy input includes both current defect state (D_t)and previously applied action (F_t) to account implicitly for unknown externalities	Dense per-step reward encouraging defect elimination: multiplicative penalty on void fraction and overprinting, with additional penalty on large action changes during overprinting to discourage oscillatory control; reward is maximised when ($[VF_{t+1}, OE_{t+1}] = [0, 0]$)	Feedforward neural network policy trained using TD3-style actor-critic optimisation via NEAT-based neuroevolution. Separate feedforward neural networks are used to learn a virtual environment mapping; defect detection CNNs and virtual environment models are part of the environment, not the RL policy/value architecture
Wang et al. [139] (2023)	Model-free MDP	High-dimensional error state derived from comparing simulated printed slice images with nominal slice images. Raw 256 × 256 error images are first processed using morphological operations and encoded via a CNN-autoencoder to extract a low-dimensional error feature vector, which is used as the RL state	Mixed discrete–continuous action space defining image-morphology-based compensation operations, including (i) discrete selection of structural element shape, (ii) discrete kernel size, and (iii) continuous iteration magnitude (positive for dilation, negative for erosion), applied at each layer	Dense per-step reward defined as an exponential function of the Dice loss between the compensated slice image and the target slice image, encouraging pixel-level error reduction	TD3 actor–critic architecture with deep neural networks: actor network based on a ResNet-style CNN, and twin critic networks implemented as fully connected networks (three layers, width 256); CNN–autoencoder used for state encoding (perception), not as the policy/value function itself
Liao et al. [140] (2023)	Model-free MDP	State defined as a 1D sliding window of future target pixels along the printing path, represented by the RGB colour values of (L) upcoming locations; the window is centred at the current nozzle position and encodes limited look-ahead to compensate for material mixing delay	Continuous action space consisting of material feed rates for each filament (CMYKW), constrained such that the total feed rate remains constant; actions are issued at each deposition step along a fixed toolpath	Dense, per-step reward defined as the negative CIELab colour difference ($-\Delta E$) between the simulated deposited material and the target image at the current location, encouraging perceptually accurate colour reproduction	Actor–critic policy representation, where both actor and critic are parameterised by convolutional neural networks (1 × 3 kernels with sinusoidal activations), trained using Proximal Policy Optimisation (PPO)
Ji et al. [141] (2022)	Model-free MDP	Explicit continuous state vector ($s_t = [\alpha_t, T_t]$), where the bending angle (α_t) is obtained from calibrated image feedback using colour-marker detection, and temperature (T_t) is estimated from the electrical resistance of the heating circuit	Continuous control action defined as the temperature change (ΔT_t), bounded by the heating system's physical limits and applied at a fixed sampling interval	Explicit asymmetric quadratic cost function penalising deviation from the reference angle, with significantly higher penalty for overshoot beyond the reference, and discounted cumulative cost minimised	Tabular Q-learning with discretisation of continuous state and action spaces into a high-resolution Q-table; no neural network or function approximation used
Ji et al. [142] (2022)	Model-free MDP	Explicit continuous state vector defined as $\left(x_t = [\alpha_t, T_t]^T\right)$, where ($\alpha_t$) is the shape angle of the SMP and (T_t) is the stimulus temperature	Explicit continuous action defined as the temperature change ($u_t = \Delta T_t$), bounded by physical heating constraints	Explicit quadratic reward function penalising deviation from the target shape angle and temperature reference as well as control effort, formulated as $\left(r_t = -w_1\widehat{\alpha_t^2} - w_2\widehat{T_t^2} - w_3 u_t^2\right)$	Q-learning with function approximation using a quadratic basis function set (Adaptive LQR via Q-learning); no neural network in the final controller
Mohammadi et al. [143] (2024)	Model-free MDP	Explicit continuous state vector composed of motor and manipulator feedback signals: rotor angle (θ), link angle (α), and their corresponding angular	Continuous control action defined as the DC motor input voltage, bounded to [−10,10] V and low-pass filtered before	Dense shaped reward combining exponential tracking error penalty relative to a reference trajectory and exponential penalty on joint oscillation	Deep actor-critic neural network architecture implemented as fully connected networks; actor and critic each use four

(*continued on next page*)

Table III (*continued*)

Author	RL Formulation	State Representation	Action Space	Reward Definition	Architecture
		velocities, obtained directly from encoder measurements	application to respect actuator constraints	amplitude, with tuneable positive coefficients	layers (input, hidden layers, output). TD3 employs twin critic networks; no convolutional or perception networks are used
Choi et al. [144] (2025)	Discrete, finite-horizon MDP	Discrete design-state encoding representing the partial configuration of a digitised design domain after placing (t) unit cells; the state implicitly encodes cell types and placement order. No sensory input; environment response is evaluated via 1D finite element analysis (FEA)	Discrete action corresponding to selecting one unit-cell type (from 12 predefined square/parallelogram cell variants) to place at the next position in a fixed tiling sequence	Sparse, terminal-only reward: zero reward for intermediate steps; final reward computed from FEA-derived performance metrics (jaw rotation angle for the gripper, horizontal/vertical latch displacement for the door latch), with additional penalisation terms for disconnected hinges	Duelling Deep Q-Network (DQN) with fully connected neural networks for Q-value approximation; five shared FC layers followed by separate value and advantage streams. No convolutional or perception networks are used
Alghamdi [145] (2025)	RL-inspired hybrid optimisation	No explicit RL state definition. The framework refers generically to "current printing parameters" and predicted performance metrics (e.g., stress, print time, material usage) produced by neural network regressors; no observation model, discretisation, or state transition structure is specified	Actions are described qualitatively as dynamic adjustments of printing parameters (e.g., layer thickness, extrusion speed), but action dimensionality, bounds, and discretisation are not explicitly defined	Reward is presented as a weighted composite objective penalising defects and inefficiency while encouraging performance improvements; the reward expression is illustrative and not tied to a specified RL update rule or policy optimisation procedure	No explicit RL policy or value function architecture is specified. Neural networks (CNN-MLP) are used for performance prediction; genetic algorithms and topology optimisation handle design optimisation. Reinforcement learning is referenced at a conceptual level without a defined learning architecture
Wang et al. [146] (2025)	Model-free episodic MDP	Discrete combinatorial state representing the set of already visited key points within a sliced layer; key points are deterministically generated from geometric preprocessing (grid points and boundary points via contour offsetting). No physical sensing; state evolves purely by marking visited nodes	Discrete action corresponding to selecting the next unvisited key point to traverse; action feasibility is constrained to prevent revisiting points and forming sub-loops	Terminal reward defined as the negative weighted objective combining total path length and cumulative turning angle, normalised by an initial solution; no intermediate rewards are used	Actor–critic policy-gradient architecture based on a pointer network with LSTM encoder–decoder and attention mechanism; critic network provides a baseline for variance reduction. Bezier-curve smoothing and multi-layer greedy connection are post-processing steps, not part of the RL policy
Tseng et al. [147] (2025)	Model-free episodic MDP	Discrete design state encoding the current soft/stiff material assignment on an N × N grid. Finite element simulation outputs (strain energy, stress, reaction force) are used internally by the environment for evaluation but are not included in the RL state. No in-process sensing or image-based observations are used	Discrete action corresponding to modifying the material distribution in the grid (material reassignment). The exact action set and granularity are not explicitly specified; feasibility constraints such as fixed soft–stiff material ratios are enforced implicitly	Reward derived from FEM-evaluated mechanical performance. Single-objective rewards maximise a selected metric (e.g., strain energy or reaction force); multi-objective rewards are defined via weighted aggregation of normalised decision values combining competing objectives (e.g., maximise strain energy while minimising stress)	Deep Q-Network with fully connected neural networks for Q-value approximation and experience replay; tabular Q-learning is replaced by neural function approximation. Network architecture details are not fully specified; FEM solvers and material models are part of the environment, not the RL architecture

Appendix D. Comparative case study: Achieving target layer geometries in thin-walled WAAM components via MIMO feedback

D.1. Introduction and motivation

In the WAAM process, thermal effects significantly affect layer geometry stability in thin-walled structures, such that fixing process parameters alone does not guarantee geometrical consistency. It has been demonstrated that, during layer-by-layer deposition, the layer width tends to increase while the layer height decreases, resulting in a progressively flatter bead profile. This behaviour is associated with a reduction in melt pool viscosity as temperature rises, which is directly observable through a decrease in the wetting angle (θ) [148] (see Fig. A1). Consequently, while it may be reasonable to assume approximately constant layer geometry during multi-bead wall deposition, this assumption does not hold for thin-walled structures, where geometry varies along the build. This variability necessitates online or intra-layer adjustment of process parameters to ensure dimensional stability.

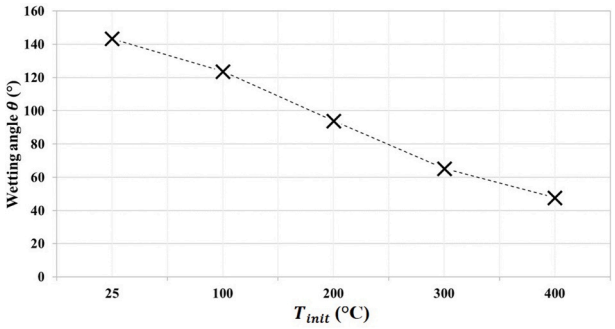

Fig. A1. From Manokruang [148], Al5356 1.2 mm wire, with CMT process with fixed wire feed speed (5 m/min) and welding speed (10 mm/s).

Despite this, the development of an effective controller is not straightforward and generally requires a MIMO control architecture. Indeed, the influence of process parameters on the WAAM process is strongly coupled and non-uniform [149], with each parameter affecting the deposition behaviour in different and often non-linear ways. More specifically, the main process parameters do not affect bead geometry in a uniform or monotonic manner.

- An increase in wire feed speed (WFS) generally leads to an increase in bead size, as higher material deposition rates tend to increase both bead height and width, depending on the local thermal conditions.
- Conversely, an increase in welding speed (WS) reduces the material deposited per unit length, typically resulting in a decrease in bead height and a reduction in overall bead volume.
- The reference welding voltage (V) exhibits a more complex and strongly non-linear influence compared to WFS and WS. Changes in voltage affect arc length, heat distribution, and droplet transfer mode, leading to non-monotonic variations in bead geometry. As a result, small voltage adjustments may produce disproportionate changes in bead width and wetting behaviour, particularly under elevated thermal conditions.
- Similarly, the contact tip to workpiece distance (CTWD) has a non-linear and coupled effect on the process. An increase in CTWD raises the electrical resistance of the arc, which tends to reduce the welding current [150]. However, at the same time, a larger CTWD increases arc length, which may promote a wider heat distribution and, under certain conditions, an increase in layer height, even in the presence of reduced current. These competing effects act in different directions, making the net influence of CTWD on bead geometry highly dependent on the thermal state of the process.
- The Gas Flow Rate (GFR) also plays a non-negligible role [151], particularly through its influence on arc stability and wetting behaviour. Variations in GFR affect the shielding effectiveness and can modify the wetting angle, thereby altering bead spreading and layer consistency.
- Finally, as previously discussed, interpass temperature [79] further amplifies these interactions by reducing melt pool viscosity and modifying solidification dynamics, leading to progressive changes in bead geometry across layers, with increment of layer width and reduction of layer height as it increases.

Overall, the combined effects of WFS, WS, voltage, CTWD, GFR, and interpass temperature are strongly coupled, non-linear, and not aligned along a single direction of influence. This complexity explains why fixed-parameter strategies are insufficient for thin-walled WAAM structures and reinforces the need for adaptive, multivariable control approaches capable of accounting for competing and state-dependent parameter effects.

In this context, the use of a conventional PID controller is inherently limited. If the objective is to regulate either bead height or bead width independently, a PID controller may be employed by fixing all other process parameters and varying a single control input. However, this approach becomes infeasible when simultaneous control of both bead height and bead width is required. Although it may be tempting to model the system as two decoupled SISO subsystems, one for layer height and one for layer width, in practice these variables are strongly coupled. As a result, independent control actions lead to conflicting responses, making SISO control strategies unsuitable. This limitation necessitates the adoption of MIMO control architectures. Within the MIMO framework, both linear and non-linear control strategies can be considered. Linear approaches include, for example, LQR, whereas non-linear approaches may involve MPC, in which the system behaviour over a finite prediction horizon is estimated using a non-linear model.

Accordingly, the primary objective of this review was to highlight the relevance of RL for control design in advanced manufacturing systems like AM. The simulated case study presented in this appendix of the work is intended not only to enable a comparison among different control strategies, but also to illustrate a systematic methodology for application development, including guidance on data acquisition requirements and model structuring.

D.2. Data collection, environment and control policy design

The experimental validation was performed using an automated welding cell (see Fig. A2) featuring an Omron Viper s850 6-axis robotic manipulator. The system was integrated with a Miller Phoenix 460 power supply, providing closed-loop regulation of welding voltage and WFS. For the deposition trials, we employed a constant voltage GMAW process using 1.0 mm diameter ER70S-6 mild steel wire with M21 shielding gas (Argon-18% CO_2) on S355JR structural steel substrates.

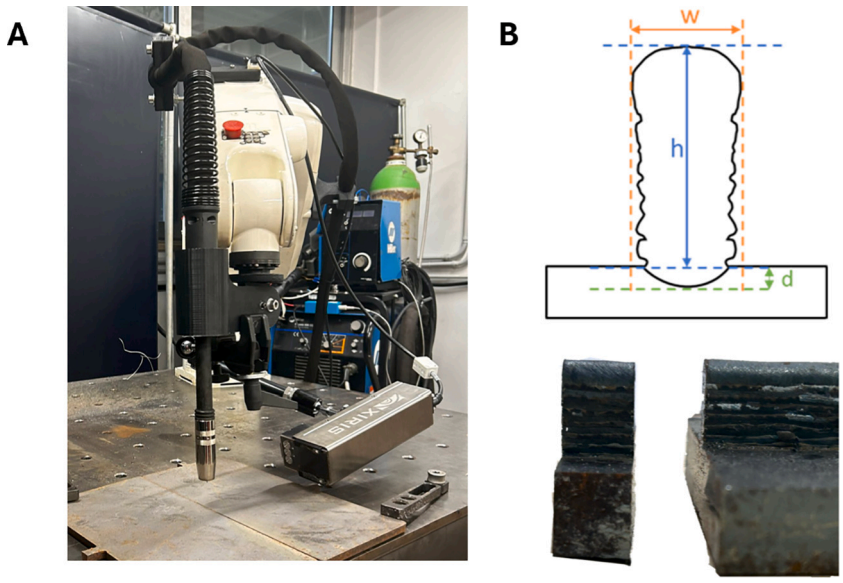

Fig. A2. The WAAM system was employed in this study. (A) The welding equipment (B) An example of the produced specimen and the measured values at the confocal microscope.

The deposition tests were conducted under varying conditions: four welding voltages (18, 20, 22, and 24 V), five wire feed speeds (2.5, 3.5, 4.5, 6, and 7 m/s), two CTWDs (12 and 20 mm), two gas flow rates (15 and 18 L/min), and three welding speeds (280, 487, and 561 mm/min). The wire was deposited onto a substrate plate, aiming to build wall structures of 10 layers each, maintaining constant process parameters throughout the tests. The deposited layer geometry was quantitatively characterised using confocal optical microscopy. Specimens were sectioned, polished, and prepared following standard metallographic procedures before measurement. The system provided high-resolution surface topography data, enabling precise extraction of steady-state bead width and height profiles across multiple deposition layers, as shown in Fig. A3.

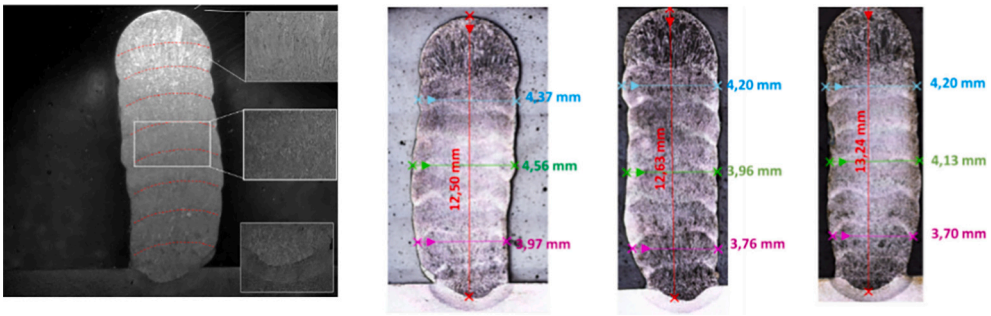

Fig. A3. A detail of the manufactured specimens observed under an optical microscope. The procedure allowed the geometry of the layers to be determined.

The collected data, obtained through destructive testing to evaluate layer geometry variations along the build direction, confirm the layer geometry variation along the building direction. As visible in Fig. A3, the layer width progressively increases along the deposition path, while the layer height decreases until thermal equilibrium is achieved. Although mean geometric values can be extracted, also the steady-state measurements (taken from stabilised layers after solidification) can be used as predictors.

However, both measurements (layer width and layer height) alone are insufficient to capture the full process dynamics. While steady-state layer geometry and process parameters can establish predictive relationships [81,92,152], such an approach neglects the inherent dynamic nature of WAAM. Previous studies have demonstrated that the geometric characteristics at any given point (g_1) depend not only on its local process parameters but also on the thermal history of preceding deposits (g_0), as formalised in the Goldak thermal model [154], shown in Fig. A4. This path-dependent behaviour underscores the need for dynamic modelling approaches that account for thermal accumulation effects throughout the deposition process. For example, Li et al. [155] proposed the usage of Long Short-Term Memory (LSTM) models to capture the dynamic nature of the process. However, for this case study a control-theoretic modelling approach is adopted. The process dynamics are approximated using a classical first-order transfer function, while the steady-state behaviour is represented either by a linear combination of the inputs or by a non-linear mapping. Experimental observations obtained using a Xiris XVC 1000 welding camera indicate that, according to literature data, the bead formation process can be characterised by a dominant time constant of approximately $\tau = 0.2\,s$.

At this point, a dynamic system that simulates the WAAM's dynamics is obtained, namely a Reduced Order Model (ROM) [156]. Accordingly, the underlying assumption is that the process parameters predominantly influence the steady-state behaviour of the system, whereas the solidification dynamics are only weakly dependent on parameter selection. For a given material, wire diameter, and operating range corresponding to the same metal transfer mode and comparable heat input, variations in the time constant are expected to be limited and not sufficient to introduce dynamic instability under control action. Moreover, uncertainties and small variations in layer geometry [157,158], typically within 10%, arising from non-linear disturbances and unmodelled factors are neglected for the purposes of this study. In this framework, the effects of lower-level control loops are embedded within the ROM, and open-loop stability is assumed.

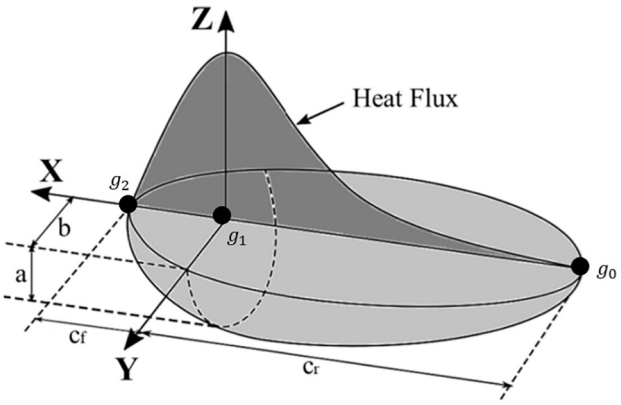

Fig. A4. The Goldak model suggests the dynamic behaviour of the arc welding process.

Two different ROMs have been developed. The MIMO linear model has been obtained via linear regression, while the non-linear model, used for the MPC and DDPG controller development, has been obtained as a Hammerstein–Wiener MIMO model [109]. From a mathematical perspective, if a first-order dynamics is assumed for both the layer width and the layer height formation process, once the steady-state value x_{SS} is determined (in a linear o nonlinear way), this value is passed as input to a space state model,

$$\begin{cases} \dot{\mathrm{x}} = -5 \cdot \mathrm{x} + \mathrm{x}_{\mathrm{ss}} \\ \mathrm{y} = 5 \cdot \mathrm{x} \end{cases} \tag{A1}$$

in which $A = {}^{1}/_{\tau} = {}^{1}/_{0.2} = 5$.

- Linear Model

A multivariate linear regression model assumes a linear relationship between the process parameters and the resulting layer geometry. Let $\mathrm{X} \in \mathbb{R}^m$ denote the vector of process parameters and $\mathrm{Y} \in \mathbb{R}^p$ the vector of layer geometry outputs. Under this assumption, the steady-state relationship can be expressed as

$$\mathrm{Y} = \mathrm{AX} + \mathrm{B} \tag{A2}$$

where $\mathrm{A} \in \mathbb{R}^{p \times m}$ is the regression coefficient matrix and $\mathrm{B} \in \mathbb{R}^p$ is a constant bias term. In the present case, the output vector is defined as

$$\mathrm{Y} = \begin{bmatrix} w \\ h \end{bmatrix},$$

while the input vector is given by

$$\mathrm{X} = [\,\mathrm{WFS} \quad \mathrm{WS} \quad \mathrm{CTWD} \quad V\,]^{\top}.$$

The linear regression framework relies on two key assumptions: (i) a linear dependence between inputs and outputs, and (ii) the absence of multicollinearity among the input parameters and output variables. While the first assumption may hold locally within specific WAAM process windows, the second assumption is frequently violated in practice due to the strong coupling between process parameters. Nevertheless, as discussed in Section 2 of the review, simplified linear models remain widely adopted in the literature primarily owing to their computational efficiency, ease of identification, and suitability for classical control design. Despite their limited ability to capture non-linear interactions and cross-parameter dependencies, linear models are a necessary prerequisite for the implementation of traditional model-based control strategies. For this reason, a data-driven linear model is identified in this study and subsequently embedded within a control-oriented dynamic framework.

The regression model is trained using 80% of the available dataset, while the remaining 20% is reserved for testing. The model parameters are estimated using the ordinary least squares (OLS) estimator, yielding the following steady-state relationships:

$$\begin{cases} h_{ss} = 0.97\,\mathrm{WFS} - 0.006\,\mathrm{WS} + 0.017\,\mathrm{CTWD} - 0.005\,V, \\ w_{ss} = 0.15\,\mathrm{WFS} - 0.005\,\mathrm{WS} - 0.038\,\mathrm{CTWD} + 0.29\,V. \end{cases} \tag{A3}$$

The mean squared error (MSE) computed on the test dataset is equal to 0.02, indicating a reasonable approximation of the steady-state geometry within the considered operating range. To incorporate transient behaviour, the steady-state model is embedded within a first-order linear dynamic system. The state vector is defined as.

$$\mathbf{x}(t) = [\,h(t) \quad w(t)\,]^{\top},$$

and its evolution is described in state-space form as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{x}_{ss}(t), \tag{A4}$$

where $\mathbf{x}_{ss}(t)$ represents the steady-state geometry predicted by the linear regression model. The system matrices are defined as

$$\mathbf{A} = \begin{bmatrix} -5 & 0 \\ 0 & -5 \end{bmatrix}, \tag{A5}$$

and

$$\mathbf{B} = 5\begin{bmatrix} 0.97 & -0.006 & 0.017 & -0.005 \\ 0.15 & -0.005 & -0.038 & 0.29 \end{bmatrix}. \tag{A6}$$

This formulation yields a stable first-order linear system with decoupled dynamics for the geometric states, while preserving the coupled steady-state dependence on the process parameters through the input matrix **B**.

- Non-linear model

As discussed, the WAAM process may exhibit both static and dynamic non-linearities arising from thermal effects, arc behaviour, and material flow phenomena. To capture these characteristics within a control-oriented framework, the non-linear plant or environment can be modelled using a Hammerstein-Wiener structure, expressed as

$$\mathbf{y}(t) = \mathscr{W}(\mathscr{G}(s)\mathscr{H}(\mathbf{u}(t))),$$

where:

- $\mathscr{H}(\cdot)$ is a static non-linear input block (Hammerstein component),
- $\mathscr{G}(s)$ is a linear time-invariant dynamic system,
- $\mathscr{W}(\cdot)$ is a static non-linear output block (Wiener component).

For this comparative case study, the WAAM process is represented by a Hammerstein-type model, in which a static non-linear block captures the steady-state mapping between the process parameters and the equilibrium bead geometry, and a shared first-order linear dynamic model governs the transient evolution of both layer height and layer width. This mapping is defined as.

$$\mathbf{z}(t) = \mathscr{H}(\mathbf{u}(t)) = f_{\boldsymbol{\theta}}(\mathbf{u}(t)),$$

where $\mathbf{z}(t)$ represents the steady-state targets of the geometric variables, and $f_{\boldsymbol{\theta}}(\cdot)$ is an unknown non-linear function parameterised by $\boldsymbol{\theta}$. This function is approximated using a shallow artificial neural network (ANN) with five normalised inputs corresponding to the process parameters, a single hidden layer of 15 neurons with hyperbolic tangent activation, and a two-neuron output layer with rectified linear unit activation representing the steady-state layer height and width. The network is trained using the RMSprop algorithm with a learning rate of 0.0001 for 2000 epochs, employing 80% of the dataset for training and 20% for testing. The resulting test MSE of 0.0018 demonstrates improved accuracy over linear regression, reflecting the non-linear dependence between process parameters and bead geometry.

As discussed in the review paper, both control-theoretic and RL approaches require the definition of a model of the process to be controlled. In classical control theory, this model is referred to as the *plant*, whereas in RL it is denoted as the environment. Despite the different terminology, both concepts represent the same underlying object, namely a simulation model of the process used for controller or policy design. In both frameworks, the accuracy of this model directly affects the quality and robustness of the resulting control strategy. In classical control, the controller is explicitly designed based on the plant model, typically by solving an optimisation problem with a closed-form or structured solution. Examples include LQR, obtained via the solution of an algebraic Riccati equation, and MPC, which relies on the repeated solution of a finite-horizon optimisation problem. In contrast, within the RL framework, the controller is replaced by a *policy* π, which maps the system state to control actions. The control actions in RL correspond directly to the manipulated inputs in classical control, while the policy is learned iteratively through interaction with the environment rather than derived from an explicit analytical formulation. Despite these differences, a strong correspondence exists between the two approaches. In particular, both frameworks rely on the optimisation of a performance criterion. In control theory, this criterion is defined through a cost function, which may include tracking errors, control effort, and additional penalty terms. In RL, an analogous role is played by the reward function, which guides the policy learning process.

For the problem considered in this work, stability is interpreted not in a strict dynamical systems sense, but as process stability, meaning the ability to maintain consistent bead geometry, thermal behaviour, and operational safety during deposition. Consequently, the reward function must account not only for setpoint tracking performance, but also for additional terms related to process stability, energy or power consumption, and constraint satisfaction. Formally, following [5], the reward at each step t can be computed as

$$R_t = \begin{cases} \Delta_2 \xi_t, & \delta_t < \epsilon \\ -\left(\delta_t^2 + k\delta + \Delta_1\right)\xi_t, & \textit{otherwise} \end{cases} \tag{A7}$$

where δ_t is the vector of tracking errors ξ_t, is a scalar term encoding process stability indicator, and $\Delta_1 < \Delta_2$, and k are weighting parameters. In this reward computation, a weighted positive reward is given if the system states (the errors) are in a region defined by ϵ, otherwise a polynomial-constrained negative reward proportional to the error is given to the agent.

With respect to constraints on the control action space, stability considerations are addressed differently in classical control and RL frameworks:

- the LQR formulation, control actions are clipped after the solution of the Riccati equation to ensure that the applied inputs remain within physically admissible process limits.

- MPC, constraints are explicitly enforced through the solution of a constrained optimisation problem, where bounds are directly imposed on the admissible values of the control inputs.
- In contrast, RL framework adopted in this work, action constraints are handled implicitly through the architecture of the policy network. Specifically, in this comparison a DDPG algorithm is employed, in which the policy maps the system state to continuous control actions. The output layer of the actor network uses a hyperbolic tangent activation function, which naturally bounds the normalised actions within a fixed range. These bounded actions are subsequently rescaled to generate admissible reference values for the low-level controllers governing the physical process variables.

A key aspect of the proposed formulation is the definition of the state vector. Rather than including only the current geometric states, namely layer height and layer width at time step *t*, the state vector is augmented with additional information, including the control action applied at the previous time step and the reference values to be tracked. As a result, the policy does not depend solely on the instantaneous system state, but also on past control actions and desired operating conditions. From a control-theoretic perspective, this formulation enables the policy to implicitly account for control rate variations and smoothness, similarly to the inclusion of control increment penalties in MPC cost functions. Moreover, the policy outputs do not act directly on the process, but instead generate setpoints for the underlying low-level controllers. This hierarchical structure improves robustness and aligns the RL formulation with established industrial control architectures.

D.3. Comparison analysis

To enable the comparison of the controllers, a step function is applied. The controller is tested by adjusting the reference point from a layer width of 6 mm and layer height of 2.5 mm to a layer width of 3.5 mm, while maintaining the same bead height. Additionally, another step response is simulated by keeping the layer width fixed at 6 mm and adjusting the layer height to 1.5 mm. Fig. A5 illustrates the step response used to test the controller's performance. The timestep of the simulation is 0.01 s, while the whole episode has a duration of 2 s, with a total of 200 interactions between the policies and the environment.

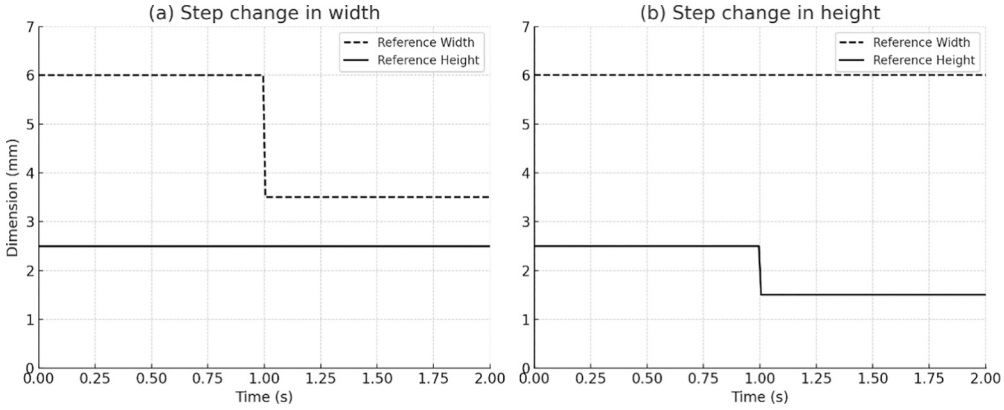

Fig. A5. Step reference changes for controller testing. (a) The reference layer width decreases from 6 mm to 3.5 mm at 1 s (b) The reference layer height decreases from 2.5 mm to 1.5 mm at 1 s.

- Performance of LQR controller

Using a value of $Q = I_{2x2}$ and $R = 0.1 \cdot I_{4x4}$ it is possible to solve the algebraic Riccati equation and obtain the optimal gain K_{LQR}, which defines the control action ($u = K_{LQR}e$). The simulation results, shown in Fig. A6, reveal poor step response performance, which is due to constraining the control variable to the minimum and maximum clipping values from the experimental campaign. In fact, a WS of 1500 mm/min, which could be an output of the model, is not feasible in practice and is therefore constrained to a maximum value of 560 mm/min. The same scaling ratio is applied to all other control variables. Based on the simulation results that include the constrained control actions, which ensure that impractical or potentially risky values for the equipment are avoided, a MAE of 1.09 mm is obtained, with an inference time of less than 1 ms. The MAE is calculated as described in Eq. (A8), which evaluates the mean error over the course of the episode of length T using the estimated geometry from the model at time t ($\widehat{w_t}, \widehat{h_t}$) and the reference values.

$$MAE = \frac{1}{T}\sum_{t=0}^{T}\frac{\left|\widehat{w_t} - w_{t,ref}\right| + \left|\widehat{h_t} - h_{t,ref}\right|}{2} \tag{A8}$$

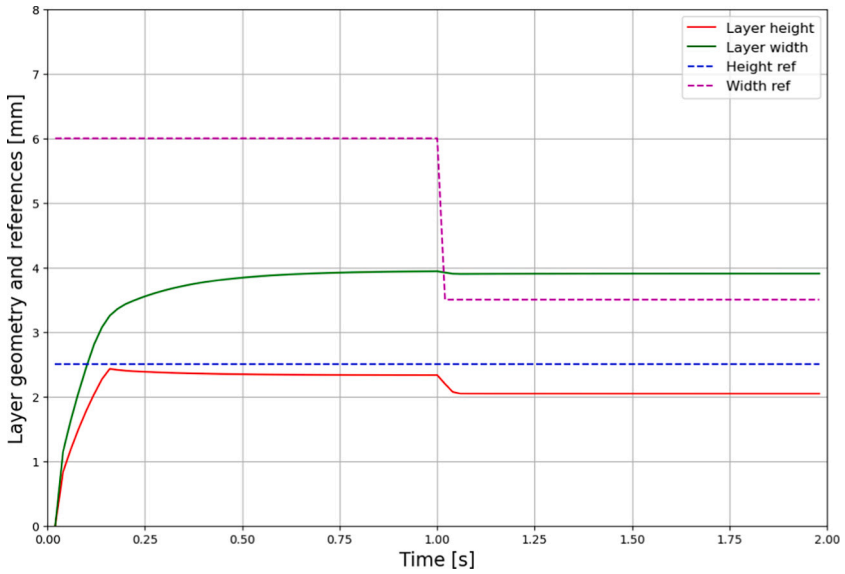

Fig. A6. Results from the simulation of the linear system plant using a MIMO Linear Quadratic Regulator (LQR) controller, obtained by clipping the control variables within the experimental range.

- Performance of MPC controller

As discussed, and demonstrated in the example provided above, modelling the WAAM process as a linear MIMO system proves to be highly inefficient. To enhance performance, a nonlinear MPC approach can be adopted, utilising the nonlinear ROM. This model is used to predict the future behaviour of the process in response to control actions. By constraining the control variables within the experimentally validated range and applying the Sequential Least Squares Programming (SLSQP) optimisation method, improved results are achieved. The outcomes shown in Fig. A7 are obtained by setting the weighting matrices to Q = [100, 100] and R = [0.1, 0.1, 0.01, 0.1, 0.1], with a prediction horizon of five steps.

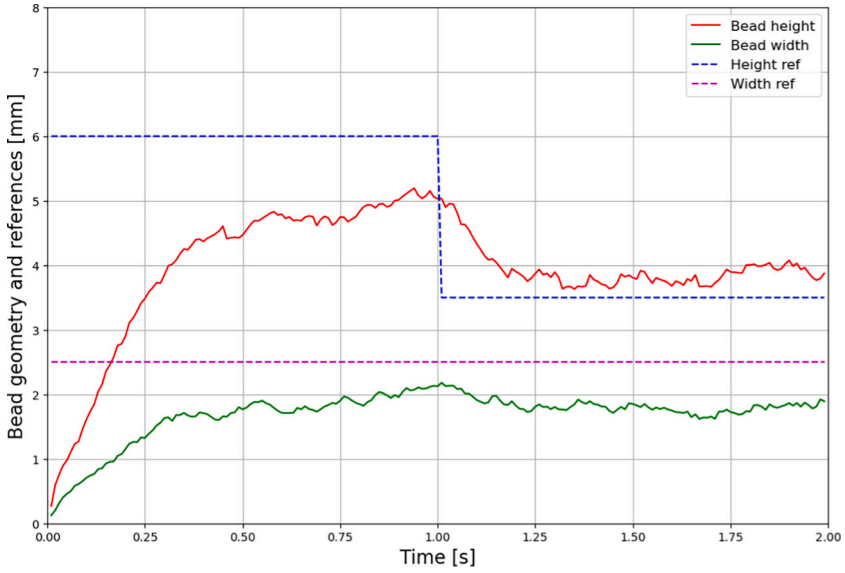

Fig. A7. Results from the simulation of the nonlinear WAAM process plant using a MIMO Model Predictive Control (MPC) controller.

Although the results obtained using the nonlinear MPC are promising, with a MAE below 1 mm, indicating that the reference setpoints can generally be reached, the practical applicability of this control strategy requires further consideration. In particular, the average computational time required to determine the optimal control action is approximately 2 s. This presents a significant limitation, as the WAAM process exhibits faster dynamics that may evolve considerably within this timeframe. Consequently, the system may not respond quickly enough to control inputs before a step change in the reference occurs, potentially leading to overshoot, instability, or delayed convergence. Therefore, while the accuracy of the predictions supports the use of the nonlinear reduced-order model, the computational latency raises concerns about real-time implementation and necessitates a deeper investigation into more efficient optimisation strategies or predictive algorithms to ensure timely control interventions.

- Performance of DDPG controller

Therefore, when considering WAAM as a nonlinear MIMO system, advanced methodologies are necessary to address the limitations of using a robust ANN for predicting future values based on input sequences. This approach, when compared to linear models, often requires more time for

computation, independent of the hardware architecture employed. In this case, the system is implemented on an AMD Ryzen 74800H 8-core processor, paired with an NVIDIA GeForce GTX 1650 Ti, which provides 4GB of dedicated memory. Therefore, a potential solution proposed in this article is the employment of RL. In this approach, the ANN used for the ROM is employed solely for control policy optimisation. Specifically, the neural network, with its optimal weights, determines the control action based on the system's states. For WAAM applications, the states considered in this study include the Mean Absolute Error (MAE) for both bead width and bead height, calculated as the difference between the reference and measured values. Additionally, the state incorporates the reference values and the previously applied action (or process parameters). These states are defined as

$$s_{WAAM,}(t) = [MAE_{,w}(t)\,, MAE_h(t)\,, r_w(t)\,, r_h(t)\,, a(t-1)\,],$$

where

- $MAE_{,w}(t)$ and $MAE_h(t)$ are the inputs for the critic and actor networks, representing the error between the reference and measured layer geometry.
- $r_w(t), r_h(t)$ is the set of reference values at time t.
- $a(t-11) = [WFS(t-1)\,, WS(t-1)\,, V(t-1)\,, CTWD(t-1)\,, GFR(t-1)\,]$ is the action or process parameters taken at the previous time step.

The critic network used in this work consists of two branches: one for states with 64 neurons and one for actions with 32 neurons, followed by a concatenation layer. Two additional layers, each with 128 neurons, are used to approximate the Q-value, once the state is given as input. ReLU activation is applied to all hidden layers, while the output layer has no activation function to estimate the Q-value. On the other hand, the actor network comprises two hidden layers, each with 64 neurons using ReLU activation, and the output layer employs a hyperbolic tangent activation function. This output layer consists of five outputs representing the process parameters to be used. The use of the hyperbolic tangent function ensures that the policy network's output remains constrained within specified minimum and maximum bounds, which are determined based on the experimental campaign. This approach helps prevent estimation issues related to bead geometry outside the training input space.

Key training settings include Adam as the optimiser, with the actor and the critic learning rates respectively of 0.0001 and 0.0002, a discount factor of 0.97, and a soft update rate of 0.01. The training procedure consists of generating a new reference point at the beginning of each episode, which ends after 200 interactions between the agent and the system, allowing for control policy generalisation in different scenarios. Finally, to allow a proper policy exploration, the white noise is reduced proceeding through the end of the training procedure, passing from active exploration at the beginning to the exploitation of the learned optimal control policy.

The result of applying the DDPG controller is shown in Fig. A8. In this case, the average computational time is 3 ms, which can be effectively used for the proposed system, given a control time lower than system time constant. Moreover, the complex control policy learnt by the agent allow to have optimal control performance with an error of 0.09 mm, considering the constrain of all the control variables. However, all the presented results are obtained in simulation, therefore, future analysis needs to be carried out to study the effective results on the real system. The final comparison of the results reached in terms of MAE during the episode and the average computational time for each policy generator is reported in Table IV.

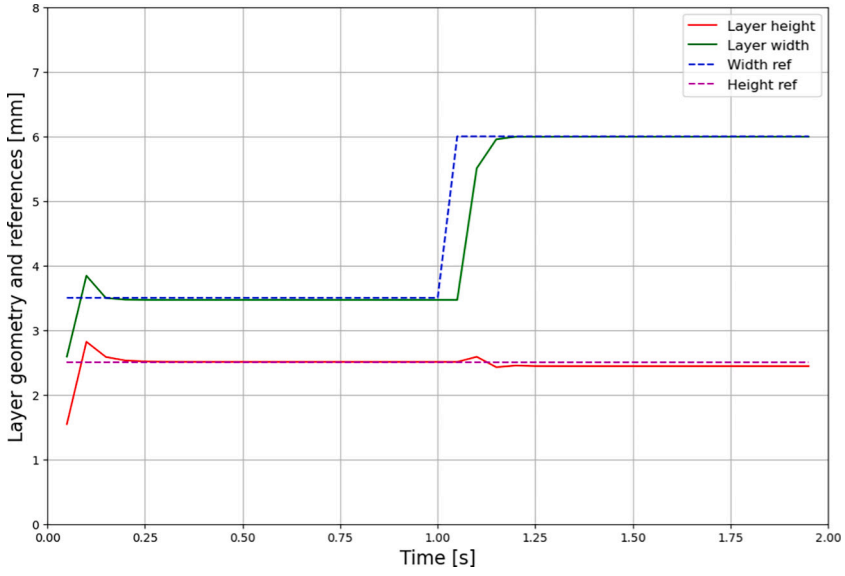

Fig. A8. Results from the simulation of the nonlinear WAAM process plant using a MIMO Deep Deterministic Policy Gradient (DDPG) controller.

Table IV
Comparison of the results.

Control law	Performance		
	Computational time	MAE on width step	High
LQR	<0.001 s	1.09 mm	1.28 mm
MPC	2 s	0.93 mm	1.1 mm
DDPG	0.003 s	0.09 mm	0.29 mm

D.4. Final discussion

This case study has been introduced as supplementary material for the main review article to provide a concrete and control-oriented illustration of the challenges discussed throughout the main paper. In particular, it reinforces the argument that classical SISO control strategies, such as PID controllers, are inadequate for AM processes characterised by strong coupling between multiple process parameters and outputs. As discussed in the review, AM processes, including WAAM, are intrinsically non-linear. The steady-state relationship between process parameters and the output quantities (e.g. layer geometry, thermal conditions, energy consumption profiles, melting pool area, etc.) is non-linear in most practical operating regimes, irrespective of the specific AM technology. Similar behaviour is well documented, for example, in LBPF, where melt pool size and geometry exhibit non-linear dependencies on laser power, scan speed, and material properties. As a consequence, linear control strategies such as linear MPC or LQR, although mathematically well established and industrially mature, are inherently limited when applied to these processes.

While the use of multivariable control architectures is necessary to address the coupled nature of the problem, adopting linear MIMO controllers does not resolve the fundamental issue associated with non-linear steady-state behaviour. Non-linear model predictive control would, in principle, represent the most rigorous solution, as it allows explicit handling of non-linear dynamics, constraints, and multivariable interactions with stability guarantees. However, this case study highlights a critical practical limitation: computational feasibility. In WAAM, the dominant process dynamics exhibit time constants on the order of several hundred milliseconds. When the control decision latency approaches or exceeds the characteristic process dynamics, real-time applicability becomes impractical. In the case of non-linear MPC relying on neural-network-based models, even a single-step prediction may require tens of milliseconds, and the computational burden increases rapidly with the prediction horizon and the number of scenarios considered. When additional delays associated with sensing, such as image acquisition and computer vision processing for layer geometry estimation, are taken into account, the overall control loop latency becomes incompatible with real-time operation.

In this context, RL emerges as a viable alternative for real-time control of complex AM processes. Unlike MPC, RL shifts the computational burden to an offline training phase, during which the policy is optimised through repeated interaction with a simulated plant. Once trained, the online execution of the policy consists of a simple forward pass through a neural network, resulting in control actions that can be computed within a few milliseconds. This property makes RL particularly attractive for processes with fast dynamics and strict real-time constraints.

The case study further demonstrates that, in RL-based control, the critical design choices are not primarily related to the depth or complexity of the neural network architecture. Rather, the formulation of the action space, the definition of the state vector, and, most importantly, the structure of the reward function plays a dominant role in determining control performance. By appropriately defining the reward, RL can implicitly account for multiple objectives, including geometric accuracy, process stability, smoothness of control actions, and energy or power consumption, without requiring explicit analytical formulations for each aspect. From a control perspective, RL can therefore be interpreted as a flexible framework capable of approximating complex non-linear control laws that would be extremely difficult to derive analytically or implement via real-time optimisation. When combined with digital twin models, RL policies can further exploit predictive information about future process evolution, reinforcing their suitability for advanced manufacturing applications.

Overall, the case study supports the central claim of the review: while classical and optimisation-based control methods remain valuable reference tools, RL represents a promising and practically scalable approach for real-time control of non-linear, multivariable AM processes, provided that careful attention is given to modelling assumptions, sensing strategies, and sim-to-real transfer. Table V summarises the key characteristics, limitations, and practical implications of the LQR, MPC, and reinforcement learning-based control strategies considered in this case study.

Table V
Concluding comparison of LQR, MPC, and RL-based control strategies for the WAAM case study.

Aspect	LQR	MPC	Reinforcement Learning (DDPG)
Process non-linearity	Not handled	Explicitly handled	Implicitly learned
MIMO capability	Yes (linear)	Yes	Yes
Constraint handling	A posteriori (clipping)	Explicit optimisation constraints	Implicit via policy architecture
Computational cost (online)	Very low	High to prohibitive	Very low
Suitability for fast dynamics	Limited	Poor (non-linear case)	High
Dependence on model accuracy	High	Very high	Moderate (robust via training)
Key design challenge	Linearisation validity	Real-time feasibility	Reward and state formulation
Industrial scalability	High (local control)	Moderate (supervisory level)	Emerging, but promising

References

[1] Xu X, Lu Y, Vogel-Heuser B, Wang L. Industry 4.0 and industry 5.0—Inception, conception and perception. J Manuf Syst Oct. 2021;61:530–5. https://doi.org/10.1016/j.jmsy.2021.10.006.

[2] Ghobakhloo M. Industry 4.0, digitization, and opportunities for sustainability. J Clean Prod Apr. 2020;252:119869. https://doi.org/10.1016/j.jclepro.2019.119869.

[4] Sousa J, et al. Artificial intelligence for control in laser-based additive manufacturing: a systematic review. IEEE Access 2025;13:30845–60. https://doi.org/10.1109/ACCESS.2025.3537859.

[5] Mattera G, Caggiano A, Nele L. Optimal data-driven control of manufacturing processes using reinforcement learning: an application to wire arc additive manufacturing. J Intell Manuf 2024;36:1291–310.

[6] Kusiak A. Smart manufacturing must embrace big data. Nature Apr. 2017;544(7648):23–5. https://doi.org/10.1038/544023a.

[7] He QP, Wang J. Statistical process monitoring as a big data analytics tool for smart manufacturing. J Process Control 2018;67:35–43.

[8] Dilberoglu UM, Gharehpapagh B, Yaman U, Dolen M. The role of additive manufacturing in the era of industry 4.0. Procedia Manuf 2017;11:545–54. https://doi.org/10.1016/j.promfg.2017.07.148.

[9] Ahmadi M, Rahmatabadi D, Karimi A, Koohpayeh MHA, Hashemi R. The role of additive manufacturing in the age of sustainable manufacturing 4.0. In: Sustainable manufacturing in industry 4.0. Singapore: Springer Nature Singapore; 2023. p. 57–78. https://doi.org/10.1007/978-981-19-7218-8_4.

[10] Sampedro GAR, Rachmawati SM, Kim D-S, Lee J-M. Exploring machine learning-based fault monitoring for polymer-based additive manufacturing: challenges and opportunities. Sensors Dec. 2022;22(23):9446. https://doi.org/10.3390/s22239446.

[11] Rajendran S, et al. A review on AI integration with FDM printing to enhance precision, efficiency, and process optimization. J Reinf Plast Compos Jul. 2025. https://doi.org/10.1177/07316844251358587.

[12] Liu J, Ye J, Silva Izquierdo D, Vinel A, Shamsaei N, Shao S. A review of machine learning techniques for process and performance optimization in laser beam powder bed fusion additive manufacturing. J Intell Manuf Dec. 2023;34(8):3249–75. https://doi.org/10.1007/s10845-022-02012-0.

[13] Zhang Y, Yan W. Applications of machine learning in metal powder-bed fusion in-process monitoring and control: status and challenges. J Intell Manuf Aug. 2023; 34(6):2557–80. https://doi.org/10.1007/s10845-022-01972-7.

[14] Moradi A, Tajalli S, Mosallanejad MH, Saboori A. Intelligent laser-based metal additive manufacturing: A review on machine learning for process optimization and property prediction. Int J Adv Manuf Technol Jan. 2025;136(2):527–60. https://doi.org/10.1007/s00170-024-14858-0.

[15] Mattera G, Nele L, Paolella D. Monitoring and control the wire arc additive manufacturing process using artificial intelligence techniques: a review. J Intell Manuf 2024;35(2):467–97.

[16] Xia C, et al. A review on wire arc additive manufacturing: monitoring, control and a framework of automated system. J Manuf Syst 2020;57:31–45.

[17] Wang L, Törngren M, Onori M. Current status and advancement of cyber-physical systems in manufacturing. J Manuf Syst Oct. 2015;37:517–27. https://doi.org/10.1016/j.jmsy.2015.04.008.

[18] Li H, Shi X, Wu B, Corradi DR, Pan Z, Li H. Wire arc additive manufacturing: a review on digital twinning and visualization process. J Manuf Process Apr. 2024; 116:293–305. https://doi.org/10.1016/j.jmapro.2024.03.001.

[19] Wei X, Wang Y, Mao J, Zhao M, Liu G. Integrating digital twin models into continuous carbon fiber-reinforced nylon additive manufacturing for process parameters verification and anomaly detection. J Intell Manuf Jun. 2025;37: 1891–908. https://doi.org/10.1007/s10845-025-02626-0.

[20] Mattera G, Vozza M, Polden J, Nele L, Pan Z. Frequency informed convolutional autoencoder for in situ anomaly detection in wire arc additive manufacturing. J Intell Manuf 2024;36:5819–34.

[21] Estalaki SM, Lough CS, Landers RG, Kinzel EC, Luo T. Predicting defects in laser powder bed fusion using in-situ thermal imaging data and machine learning. Addit Manuf Oct. 2022;58:103008. https://doi.org/10.1016/J.ADDMA.2022.103008.

[22] Mattera G, Yap EW, Polden J, Brown E, Nele L, Van Duin S. Utilising unsupervised machine learning and IoT for cost-effective anomaly detection in multi-layer wire arc additive manufacturing. Int J Adv Manuf Technol 2024;135:2957–74.

[23] Reisch R, Hauser T, Lutz B, Pantano M, Kamps T, Knoll A. Distance-based multivariate anomaly detection in wire arc additive manufacturing. In: 2020 19th IEEE international conference on machine learning and applications (ICMLA); 2020. p. 659–64.

[24] Siraskar R, Kumar S, Patil S, Bongale A, Kotecha K. Reinforcement learning for predictive maintenance: a systematic technical review. Artif Intell Rev 2023;56 (11):12885–947. https://doi.org/10.1007/s10462-023-10468-6.

[25] Zonta T, da Costa CA, da Rosa Righi R, de Lima MJ, da Trindade ES, Li GP. Predictive maintenance in the industry 4.0: a systematic literature review. Comput Ind Eng Dec. 2020;150:106889. https://doi.org/10.1016/j.cie.2020.106889.

[26] Li C, Zheng P, Yin Y, Wang B, Wang L. Deep reinforcement learning in smart manufacturing: a review and prospects. CIRP J Manuf Sci Technol Feb. 2023;40: 75–101. https://doi.org/10.1016/j.cirpj.2022.11.003.

[27] Zhang C, Juraschek M, Herrmann C. Deep reinforcement learning-based dynamic scheduling for resilient and sustainable manufacturing: a systematic review. J Manuf Syst Dec. 2024;77:962–89. https://doi.org/10.1016/j.jmsy.2024.10.026.

[29] Locatelli A, Sieniutycz S. Optimal control: an introduction. Appl Mech Rev May 2002;55(3):B48–9. https://doi.org/10.1115/1.1470675.

[30] Nele L, Mattera G, Yap EW, Vozza M, Vespoli S. Towards the application of machine learning in digital twin technology: a multi-scale review. Discov Appl Sci 2024;6(10):1–23.

[31] Chen H, Lv F, Lin T, Chen S. Closed-loop control of robotic arc welding system with full-penetration monitoring. J Intell Robot Syst Dec. 2009;56(5):565–78. https://doi.org/10.1007/s10846-009-9329-7.

[32] Zarghoon S, et al. Full-state feedback LQR with integral gain for control of induction heating of steel billet. Eng Sci Technol Int J Jul. 2024;55:101721. https://doi.org/10.1016/j.jestch.2024.101721.

[33] Anzehaee MM, Haeri M. Welding current and arc voltage control in a GMAW process using ARMarkov based MPC. Control Eng Pract 2011;19(12):1408–22.

[34] Piche S, Sayyar-Rodsari B, Johnson D, Gerules M. Nonlinear model predictive control using neural networks. IEEE Control Syst Mag 2000;20(3):53–62.

[35] Rossiter JA. Model-based predictive control. CRC Press; 2017. https://doi.org/10.1201/9781315272610.

[36] Wang K, Zhang S, Gros S, Raković SV. Tube MPC with time-varying cross-sections. IEEE Trans Automat Contr Mar. 2025;70(3):1851–8. https://doi.org/10.1109/TAC.2024.3468093.

[37] Schwenzer M, Ay M, Bergs T, Abel D. Review on model predictive control: an engineering perspective. Int J Adv Manuf Technol Nov. 2021;117(5–6):1327–49. https://doi.org/10.1007/s00170-021-07682-3.

[38] Bwambale E, et al. A review of model predictive control in precision agriculture. Smart Agric Technol Mar. 2025;10:100716. https://doi.org/10.1016/j.atech.2024.100716.

[39] Roque P, Bin E, Miraldo P, Dimarogonas DV. Fast model predictive image-based visual servoing for quadrotors. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE; Oct. 2020. p. 7566–72. https://doi.org/10.1109/IROS45743.2020.9340759.

[40] Nguyen K, Schoedel S, Alavilli A, Plancher B, Manchester Z. TinyMPC: model-predictive control on resource-constrained microcontrollers. In: 2024 IEEE international conference on robotics and automation (ICRA). IEEE; May 2024. p. 1–7. https://doi.org/10.1109/ICRA57147.2024.10610987.

[41] O'Donoghue B, Stathopoulos G, Boyd S. A splitting method for optimal control. IEEE Trans Control Syst Technol Nov. 2013;21(6):2432–42. https://doi.org/10.1109/TCST.2012.2231960.

[42] AlQahtani NA, Rogers TJ, Sims ND. Control of flexible structures using model predictive control and gaussian processes. J Phys Conf Ser Jun. 2024;2647(3): 032002. https://doi.org/10.1088/1742-6596/2647/3/032002.

[43] Kamthe S, Deisenroth MP. Data-efficient reinforcement learning with probabilistic model predictive control. In: Proceedings of the twenty-first international conference on artificial intelligence and statistics; 2018. p. 1701–10.

[44] Liu W, Wang G, Sun J, Bullo F, Chen J. Learning robust data-based LQG controllers from noisy data. IEEE Trans Automat Contr Dec. 2024;69(12): 8526–38. https://doi.org/10.1109/TAC.2024.3409749.

[45] Song Y, Scaramuzza D. Policy search for model predictive control with application to agile drone flight. IEEE Trans Robot Aug. 2022;38(4):2114–30. https://doi.org/10.1109/TRO.2022.3141602.

[46] Reinhardt D, Baumgärtner K, Frey J, Diehl M, Gros S. MPC4RL - a software package for reinforcement learning based on model predictive control. In: 2024 IEEE 63rd conference on decision and control (CDC). IEEE; Dec. 2024. p. 1787–94. https://doi.org/10.1109/CDC56724.2024.10886483.

[47] Lavretsky E, Wise KA. Robust and adaptive control. Cham: Springer International Publishing; 2024. https://doi.org/10.1007/978-3-031-38314-4.

[48] Dimitri P Bertsekas. Dynamic programming and optimal control. 3rd ed.vol. 1. Athena Scientific; 2017.

[49] Meyn Sean, Systems Control, Learning Reinforcement. 1st ed. Cambridge University Press; 2022.

[50] Yap CY, et al. Review of selective laser melting: materials and applications. Appl Phys Rev Dec. 2015;2(4). https://doi.org/10.1063/1.4935926.

[51] Svetlizky D, et al. Laser-based directed energy deposition (DED-LB) of advanced materials. Mater Sci Eng A Apr. 2022;840:142967. https://doi.org/10.1016/j.msea.2022.142967.

[52] Zapata A, Bernauer C, Stadter C, Kolb CG, Zaeh MF. Investigation on the cause-effect relationships between the process parameters and the resulting geometric properties for wire-based coaxial laser metal deposition. Metals (Basel) Mar. 2022;12(3):455. https://doi.org/10.3390/met12030455.

[53] Whip B, Sheridan L, Gockel J. The effect of primary processing parameters on surface roughness in laser powder bed additive manufacturing. Int J Adv Manuf Technol Aug. 2019;103(9–12):4411–22. https://doi.org/10.1007/s00170-019-03716-z.

[54] Zhong C, Biermann T, Gasser A, Poprawe R. Experimental study of effects of main process parameters on porosity, track geometry, deposition rate, and powder efficiency for high deposition rate laser metal deposition. J Laser Appl Nov. 2015; 27(4). https://doi.org/10.2351/1.4923335.

[55] Yuan K, et al. Influence of process parameters and heat treatments on the microstructures and dynamic mechanical behaviors of inconel 718 superalloy manufactured by laser metal deposition. Mater Sci Eng A Apr. 2018;721:215–25. https://doi.org/10.1016/j.msea.2018.02.014.

[56] Shamsaei N, Yadollahi A, Bian L, Thompson SM. An overview of direct laser deposition for additive manufacturing; Part II: mechanical behavior, process parameter optimization and control. Addit Manuf Oct. 2015;8:12–35. https://doi.org/10.1016/j.addma.2015.07.002.

[57] Hussain SZ, et al. Feedback control of melt pool area in selective laser melting additive manufacturing process. Processes Aug. 2021;9(9):1547. https://doi.org/10.3390/pr9091547.

[58] Liu B, et al. Real-time closed-loop control of molten pool transient area in direct laser deposition via PID algorithm with enhanced robustness. Int J Adv Manuf Technol Feb. 2024;130(9–10):4529–42. https://doi.org/10.1007/s00170-024-13002-2.

[59] Akbari M, Kovacevic R. Closed loop control of melt pool width in robotized laser powder–directed energy deposition process. Int J Adv Manuf Technol Oct. 2019; 104(5–8):2887–98. https://doi.org/10.1007/s00170-019-04195-y.

[60] Shi T, Shi J, Xia Z, Lu B, Shi S, Fu G. Precise control of variable-height laser metal deposition using a height memory strategy. J Manuf Process Sep. 2020;57: 222–32. https://doi.org/10.1016/j.jmapro.2020.05.026.

[61] Shi T, Lu B, Shen T, Zhang R, Shi S, Fu G. Closed-loop control of variable width deposition in laser metal deposition. Int J Adv Manuf Technol Aug. 2018;97 (9–12):4167–78. https://doi.org/10.1007/s00170-018-1895-z.

[62] Bernauer C, et al. Segmentation-based closed-loop layer height control for enhancing stability and dimensional accuracy in wire-based laser metal deposition. Robot Comput Integr Manuf Apr. 2024;86:102683. https://doi.org/10.1016/j.rcim.2023.102683.

[63] Liu Y, Wang L, Brandt M. Model predictive control of laser metal deposition. Int J Adv Manuf Technol Nov. 2019;105(1–4):1055–67. https://doi.org/10.1007/s00170-019-04279-9.

[64] Mhmood TR, Al-Karkhi NK. A review of the stereo lithography 3D printing process and the effect of parameters on quality. Al-Khwarizmi Eng J Jun. 2023;19 (2):82–94. https://doi.org/10.22153/kej.2023.04.003.

[65] Ladani L, Sadeghilaridjani M. Review of powder bed fusion additive manufacturing for metals. Metals (Basel) Sep. 2021;11(9):1391. https://doi.org/10.3390/met11091391.

[66] Osipovich K, et al. Wire-feed electron beam additive manufacturing: a review. Metals (Basel) Jan. 2023;13(2):279. https://doi.org/10.3390/met13020279.

[67] Troise M, Krichel T, Olschok S, Reisgen U. Investigation of the influence of pulse parameters on the resulting weld seam quality in pulsed electron beam welding of AW 6061. J Adv Join Process Jun. 2024;9:100183. https://doi.org/10.1016/J.JAJP.2023.100183.

[68] Vasileska E, Demir AG, Colosimo BM, Previtali B. A novel paradigm for feedback control in LPBF: layer-wise correction for overhang structures. Adv Manuf Jun. 2022;10(2):326–44. https://doi.org/10.1007/s40436-021-00379-6.

[69] Liang Z, et al. Improving process stability of electron beam directed energy deposition by closed-loop control of molten pool. Addit Manuf Jun. 2023;72:103638. https://doi.org/10.1016/J.ADDMA.2023.103638.
[70] Wickramasinghe S, Do T, Tran P. FDM-based 3D printing of polymer and associated composite: a review on mechanical properties, defects and treatments. Polymers (Basel) Jul. 2020;12(7):1529. https://doi.org/10.3390/polym12071529.
[71] Lhachemi H, Malik A, Shorten R. Augmented reality, cyber-physical systems, and feedback control for additive manufacturing: a review. IEEE Access 2019;7:50119–35. https://doi.org/10.1109/ACCESS.2019.2907287.
[72] Costin MH, Taylor PA, Wright JD. On the dynamics and control of a plasticating extruder. Polym Eng Sci Dec. 1982;22(17):1095–106. https://doi.org/10.1002/pen.760221707.
[73] Jiang J, Wen S, Zhao G. A melt temperature PID controller based on RBF neural network. In: 2008 ISECS international colloquium on computing, communication, control, and management; 2008. p. 172–5. https://doi.org/10.1109/CCCM.2008.141.
[74] Chi X, et al. Machine learning-based online monitoring and closed-loop controlling for 3D printing of continuous fiber-reinforced composites. Addit Manuf Front Jun. 2025;4(2):200196. https://doi.org/10.1016/j.amf.2025.200196.
[75] Gao H, et al. Towards quality controllable strategies in wire-arc directed energy deposition. Aug. 01, 2025, Institute of Physics Int J Extrem Manuf 2025;7:042004. https://doi.org/10.1088/2631-7990/adb9a9.
[76] Williams SW, Martina F, Addison AC, Ding J, Pardal G, Colegrove P. Wire + arc additive manufacturing. Mater Sci Technol 2016;32(7):641–7.
[77] Pan Z, Ding D, Wu B, Cuiuri D, Li H, Norrish J. Arc welding processes for additive manufacturing: a review. Trans Intell Weld Manuf 2018;I(1 2017):3–24.
[78] Wu B, et al. A review of the wire arc additive manufacturing of metals: properties, defects and quality improvement. J Manuf Process 2018;35:127–39.
[79] Kozamernik N, Bračun D, Klobčar D. WAAM system with interpass temperature control and forced cooling for near-net-shape printing of small metal components. Int J Adv Manuf Technol Sep. 2020;110(7–8):1955–68. https://doi.org/10.1007/s00170-020-05958-8.
[80] Li T, Cao Y, Ye Q, Zhang YM. Generative adversarial networks (GAN) model for dynamically adjusted weld pool image toward human-based model predictive control (MPC). J Manuf Process May 2025;141:210–21. https://doi.org/10.1016/J.JMAPRO.2025.02.053.
[81] Xiong J, Zhang G, Hu J, Wu L. Bead geometry prediction for robotic GMAW-based rapid manufacturing through a neural network and a second-order regression analysis. J Intell Manuf 2014;25(1):157–63.
[82] Mattera G, Piscopo G, Longobardi M, Giacalone M, Nele L. Improving the interpretability of data-driven models for additive manufacturing processes using clusterwise regression. Mathematics Aug. 2024;12(16):2559. https://doi.org/10.3390/math12162559.
[83] Teng S, Dehgahi S, Henein H, Wolfe T, Qureshi AJ. Sensor-fusion enabled inter-layer temperature control of nano-treated 7075 aluminum alloy produced through wire-arc directed energy deposition process. Prog Addit Manuf Aug. 2024;10:1293–314. https://doi.org/10.1007/s40964-024-00707-9.
[84] Mattera G, Polden J, Caggiano A, Nele L, Pan Z, Norrish J. Semi-supervised learning for real-time anomaly detection in pulsed transfer wire arc additive manufacturing. J Manuf Process Oct. 2024;128:84–97. https://doi.org/10.1016/j.jmapro.2024.07.142.
[85] Bellamkonda PN, Dwivedy M, Addanki R. Cold metal transfer technology - a review of recent research developments. Results Eng Sep. 2024;23:102423. https://doi.org/10.1016/j.rineng.2024.102423.
[86] Xiong J, Liu G, Pi Y. Increasing stability in robotic GTA-based additive manufacturing through optical measurement and feedback control. Robot Comput Integr Manuf Oct. 2019;59:385–93. https://doi.org/10.1016/J.RCIM.2019.05.012.
[87] Xia C, et al. Model predictive control of layer width in wire arc additive manufacturing. J Manuf Process 2020;58:179–86.
[88] Wang Y, et al. Coordinated monitoring and control method of deposited layer width and reinforcement in WAAM process. J Manuf Process 2021;71:306–16. https://doi.org/10.1016/j.jmapro.2021.09.033.
[89] Mu H, Polden J, Li Y, He F, Xia C, Pan Z. Layer-by-layer model-based adaptive control for wire arc additive manufacturing of thin-wall structures. J Intell Manuf Apr. 2022;33(4):1165–80. https://doi.org/10.1007/s10845-022-01920-5.
[90] Mu H, He F, Yuan L, Commins P, Xu J, Pan Z. High-frequency real-time bead geometry measurement in wire arc additive manufacturing based on welding signals. IEEE Trans Industr Inform Mar. 2025;21(3):2630–9. https://doi.org/10.1109/TII.2024.3514121.
[91] Chabot A, Rauch M, Hascoët J-Y. Novel control model of contact-tip-to-work distance (CTWD) for sound monitoring of arc-based DED processes based on spectral analysis. Int J Adv Manuf Technol Oct. 2021;116(11–12):3463–72. https://doi.org/10.1007/s00170-021-07621-2.
[92] Mattera G, Caggiano A, Nele L. Energy efficiency optimisation in wire arc additive manufacturing of invar 36 alloy via intelligent data-driven techniques. Int J Precis Eng Manuf-Green Technol Feb. 2025;12:905–17. https://doi.org/10.1007/s40684-025-00705-4.
[93] Kumar A, Sadhya S, Khan AU, Madhukar YK. Improvement in process efficiency of WAAM-TIG by in-situ voltage–current–temperature monitoring and feedback control system. Int J Precis Eng Manuf Jul. 2025;26(7):1673–81. https://doi.org/10.1007/s12541-024-01208-z.
[94] Sutton RS, Barto A. Reinforcement learning. In: Adaptive computation and machine learning. 2nd ed. Cambridge, Massachusetts: The MIT Press; 2018.
[95] Zhang S, Sutton RS. A deeper look at experience replay. Apr. 2018. https://doi.org/10.48550/arXiv.1712.01275.
[96] Wang Y-H, Li T-HS, Lin C-J. Backward Q-learning: the combination of Sarsa algorithm and Q-learning. Eng Appl Artif Intel 2013;26(9):2184–93.
[97] Zhao Y, Wang Y, Tan Y, Zhang J, Yu H. Dynamic jobshop scheduling algorithm based on deep Q network. IEEE Access 2021;9:122995–3011. https://doi.org/10.1109/ACCESS.2021.3110242.
[98] Vespoli S, Mattera G, Marchesano MG, Nele L, Guizzi G. Adaptive manufacturing control with deep reinforcement learning for dynamic WIP management in industry 4.0. Comput Ind Eng Apr. 2025;202:110966. https://doi.org/10.1016/J.CIE.2025.110966.
[99] Canzini E, Auledas-Noguera M, Pope S, Tiwari A. Decision making for multi-robot fixture planning using multi-agent reinforcement learning. IEEE Trans Autom Sci Eng 2025;22:5578–89. https://doi.org/10.1109/TASE.2024.3424677.
[100] Mnih V, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33.
[101] Kakade S, Langford J. Approximately optimal approximate reinforcement learning. In: Proc. 19th international conference on machine learning; 2002.
[102] Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. In: International conference on machine learning; 2015. p. 1889–97.
[103] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. 2017.
[104] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. 2018. https://doi.org/10.48550/arXiv.1801.01290.
[105] Dankwa S, Zheng W. Twin-delayed DDPG. In: Proceedings of the 3rd international conference on vision, image and signal processing. New York, NY, USA: ACM; Aug. 2019. p. 1–5. https://doi.org/10.1145/3387168.3387199.
[106] Zhang T, Yuan C, Zou Y. Research on the algorithm of constant force grinding controller based on reinforcement learning PPO. Int J Adv Manuf Technol Jun. 2023;126(7–8):2975–88. https://doi.org/10.1007/s00170-023-11129-2.
[107] Hao X, et al. Deep reinforcement learning enhanced PID control for hydraulic servo systems in injection molding machines. Sci Rep Jul. 2025;15(1):23005. https://doi.org/10.1038/s41598-025-05904-2.
[108] Zhang Y, Zhu H, Tang D, Zhou T, Gui Y. Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems. Robot Comput Integr Manuf Dec. 2022;78:102412. https://doi.org/10.1016/J.RCIM.2022.102412.
[109] Bloemen HHJ, Van Den Boom TJJ, Verbruggen HB. Model-based predictive control for Hammerstein?Wiener systems. Int J Control Jan. 2001;74(5):482–95. https://doi.org/10.1080/00207170010014061.
[110] Wasmer K, Le-Quang T, Meylan B, Shevchik SA. In situ quality monitoring in AM using acoustic emission: a reinforcement learning approach. J Mater Eng Perform Feb. 2019;28(2):666–72. https://doi.org/10.1007/s11665-018-3690-2.
[111] Venugopal V, Anand S. Structural and thermal generative design using reinforcement learning-based search strategy for additive manufacturing. Manuf Lett Aug. 2023;35:564–75. https://doi.org/10.1016/j.mfglet.2023.08.030.
[112] Huang Y, et al. Learning based toolpath planner on diverse graphs for 3D printing. ACM Trans Graph Dec. 2024;43(6):1–16. https://doi.org/10.1145/3687933.
[113] Qin M, Ding J, Qu S, Song X, Wang CCL, Liao W-H. Deep reinforcement learning based toolpath generation for thermal uniformity in laser powder bed fusion process. Addit Manuf Jan. 2024;79:103937. https://doi.org/10.1016/j.addma.2023.103937.
[114] Vagenas S, Al-Saadi T, Panoutsos G. Multi-layer process control in selective laser melting: a reinforcement learning approach. J Intell Manuf Dec. 2024;37:281–98. https://doi.org/10.1007/s10845-024-02548-3.
[115] Park B, Chen A, Mishra S. Real-time melt pool homogenization through geometry-informed control in laser powder bed fusion using reinforcement learning. IEEE Trans Autom Sci Eng 2025;22:2986–97. https://doi.org/10.1109/TASE.2024.3386882.
[116] Faizan Mohamed AM, Careri F, Khan RHU, Attallah MM, Stella L. A novel porosity prediction framework based on reinforcement learning for process parameter optimization in additive manufacturing. Scr Mater Jan. 2025;255:116377. https://doi.org/10.1016/j.scriptamat.2024.116377.
[117] Yin J, Ji X, Liang SY. Process planning for molten pool stabilization of laser powder bed fusion. Opt Laser Technol Oct. 2025;188:112983. https://doi.org/10.1016/j.optlastec.2025.112983.
[118] Knaak C, Masseling L, Duong E, Abels P, Gillner A. Improving build quality in laser powder bed fusion using high dynamic range imaging and model-based reinforcement learning. IEEE Access 2021;9:55214–31. https://doi.org/10.1109/ACCESS.2021.3067302.
[119] Malik AW, Mahmood MA, Liou F. Digital twin–driven optimization of laser powder bed fusion processes: a focus on lack-of-fusion defects. Rapid Prototyp J Nov. 2024;30(10):1977–88. https://doi.org/10.1108/RPJ-02-2024-0091.
[120] Ogoke F, Farimani AB. Thermal control of laser powder bed fusion using deep reinforcement learning. Addit Manuf Oct. 2021;46:102033. https://doi.org/10.1016/j.addma.2021.102033.
[121] Grais EM, Notley SV, Panoutsos G. Reinforcement learning for multiple-input multiple-output control in metal additive manufacturing. In: 2023 IEEE international conference on networking, sensing and control (ICNSC). IEEE; Oct. 2023. p. 1–6. https://doi.org/10.1109/ICNSC58704.2023.10319015.
[122] Vagenas S, Panoutsos G. Stability in reinforcement learning process control for additive manufacturing. IFAC-PapersOnLine 2023;56(2):4719–24. https://doi.org/10.1016/j.ifacol.2023.10.1233.

[123] Dharmadhikari S, Menon N, Basak A. A reinforcement learning approach for process parameter optimization in additive manufacturing. Addit Manuf Jun. 2023;71:103556. https://doi.org/10.1016/j.addma.2023.103556.
[124] Petrik J, Bambach M. RLTube: reinforcement learning based deposition path planner for thin-walled bent tubes with optionally varying diameter manufactured by wire-arc additive manufacturing. Manuf Lett Jul. 2024;40:31–6. https://doi.org/10.1016/j.mfglet.2024.01.007.
[125] Petrik J, Bambach M. Reinforcement learning and optimization based path planning for thin-walled structures in wire arc additive manufacturing. J Manuf Process May 2023;93:75–89. https://doi.org/10.1016/j.jmapro.2023.03.013.
[126] Sideris I, Petrik J, Bambach M. Too hot to print, too slow to handle; finding optimal path characteristics for WAAM. Manuf Lett Oct. 2024;41:879–90. https://doi.org/10.1016/j.mfglet.2024.09.108.
[127] Ferreira RP, Schubert E, Scotti A. Exploring multi-armed bandit (MAB) as an AI tool for optimising GMA-WAAM path planning. J Manuf Mater Process May 2024;8(3):99. https://doi.org/10.3390/jmmp8030099.
[128] Ferreira RP, Schubert E, Scotti A. Reducing computational time in pixel-based path planning for GMA-DED by using multi-armed bandit reinforcement learning algorithm. J Manuf Mater Process Mar. 2025;9(4):107. https://doi.org/10.3390/jmmp9040107.
[129] Li Z, Segura LJ, Li Y, Zhou C, Sun H. Multiclass reinforced active learning for droplet pinch-off behaviors identification in inkjet printing. J Manuf Sci Eng Jul. 2023;145(7). https://doi.org/10.1115/1.4057002.
[130] Dharmawan AG, Xiong Y, Foong S, Song Soh G. 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). IEEE; May 2020. p. 4030–6. https://doi.org/10.1109/ICRA40945.2020.9197222.
[131] Xue B, et al. Intelligent monitoring and control system for molten metal drop-on-demand jetting by water-hammer effect. Precis Eng Oct. 2025;96:134–46. https://doi.org/10.1016/j.precisioneng.2025.06.010.
[132] Patrick S, Nycz A, Noakes M. Reinforcement learning for generating toolpaths in additive manufacturing. University of Texas at Austin; 2018.
[133] Ge J, et al. A reinforcement learning-based path planning method for complex thin-walled structures in 3D printing. In: 2021 the 5th international conference on innovation in artificial intelligence. New York, NY, USA: ACM; Mar. 2021. p. 234–40. https://doi.org/10.1145/3461353.3461382.
[134] Piovarči M, et al. Closed-loop control of direct ink writing via reinforcement learning. ACM Trans Graph Jul. 2022;41(4):1–10. https://doi.org/10.1145/3528223.3530144.
[135] Mishra A, Jatti VS. Reinforcement learning based approach for the optimization of mechanical properties of additively manufactured specimens. Int J Interact Des Manuf. (IJIDeM) Aug. 2023;17(4):2045–53. https://doi.org/10.1007/s12008-023-01257-0.
[136] Zavrakli E, Parnell A, Dey S. Reinforcement learning based temperature regulation for a material extrusion additive manufacturing system. 2023.
[137] Chung J, Shen B, Law ACC, Kong Z (James). Reinforcement learning-based defect mitigation for quality assurance of additive manufacturing. J Manuf Syst Oct. 2022;65:822–35. https://doi.org/10.1016/j.jmsy.2022.11.008.
[138] Cleeman J, Jackson A, Esola S, Shao C, Xu H, Malhotra R. Scalable control of extraneously induced defects in in-field additive manufacturing. J Manuf Process May 2025;141:919–33. https://doi.org/10.1016/j.jmapro.2025.03.014.
[139] Wang D, et al. Deep reinforcement learning for dynamic error compensation in 3D printing. In: 2023 IEEE 19th international conference on automation science and engineering (CASE). IEEE; Aug. 2023. p. 1–7. https://doi.org/10.1109/CASE56687.2023.10260588.
[140] Liao K, Tricard T, Piovarči M, Seidel H-P, Babaei V. Learning deposition policies for fused multi-material 3D printing. In: 2023 IEEE international conference on robotics and automation (ICRA). IEEE; May 2023. p. 12345–52. https://doi.org/10.1109/ICRA48891.2023.10160465.
[141] Ji Q, Chen M, Wang XV, Wang L, Feng L. Optimal shape morphing control of 4D printed shape memory polymer based on reinforcement learning. Robot Comput Integr Manuf Feb. 2022;73:102209. https://doi.org/10.1016/j.rcim.2021.102209.
[142] Ji Q, Wang XV, Wang L, Feng L. Online reinforcement learning for the shape morphing adaptive control of 4D printed shape memory polymer. Control Eng Pract Sep. 2022;126:105257. https://doi.org/10.1016/j.conengprac.2022.105257.
[143] Mohammadi M, et al. Sustainable robotic joints 4D printing with variable stiffness using reinforcement learning. Robot Comput Integr Manuf Feb. 2024;85:102636. https://doi.org/10.1016/j.rcim.2023.102636.
[144] Choi Y, Kim Y, Park K. Deep reinforcement learning for optimal design of compliant mechanisms based on digitized cell structures. Eng Appl Artif Intel Jul. 2025;151:110702. https://doi.org/10.1016/j.engappai.2025.110702.
[145] Alghamdi A. Enhancing 3D printing workflows through multi-objective optimization and reinforcement learning techniques. Eng Technol Appl Sci Res Apr. 2025;15(2):21300–5. https://doi.org/10.48084/etasr.10101.
[146] Wang X, et al. Reinforcement learning-based continuous path planning and automated concrete 3D printing of complex hollow components. Autom Constr Sep. 2025;177:106290. https://doi.org/10.1016/j.autcon.2025.106290.
[147] Tseng B, et al. Optimizing bioinspired composite materials using deep Q-network reinforcement learning and finite element methods. Adv Eng Mater Apr. 2025;27(22):2402807. https://doi.org/10.1002/adem.202402807.
[148] Manokruang S. Phenomenological model of thermal effects on weld beads geometry produced by Wire and Arc Additive Manufacturing (WAAM)', Université Grenoble Alpes [2020–…] [Online]. Available: https://theses.hal.science/tel-03851334; 2022.
[149] Vozza M, Polden J, Mattera G, Piscopo G, Vespoli S, Nele L. Explaining the anomaly detection in additive manufacturing via boosting models and frequency analysis. Mathematics Oct. 2024;12(21):3414. https://doi.org/10.3390/math12213414.
[150] Henckell P, Gierth M, Ali Y, Reimann J, Bergmann JP. Reduction of energy input in wire arc additive manufacturing (WAAM) with gas metal arc welding (GMAW). Materials May 2020;13(11):2491. https://doi.org/10.3390/ma13112491.
[151] Pires JN, Loureiro A, Bölmsjo G. Welding robots: technology, system issues and application. Springer Science & Business Media; 2006.
[152] Ding D, He F, Yuan L, Pan Z, Wang L, Ros M. The first step towards intelligent wire arc additive manufacturing: An automatic bead modelling system using machine learning through industrial information integration. J Ind Inf Integr Sep. 2021;23:100218. https://doi.org/10.1016/j.jii.2021.100218.
[154] Goldak JA, Akhlaghi M. Computational welding mechanics. Springer Science & Business Media; 2005.
[155] Li Z, Hou Z, Pan Z, Wu D, Xu J. A non-autoregressive dynamic model based welding parameter planning method for varying geometry beads in WAAM. IEEE Trans Ind Electron 2022;70(3):2770–9.
[156] Xiao J, Liu N, Lua J, Saathoff C, Seneviratne W p. Data-driven and reduced-order modeling of composite drilling. In: AIAA scitech 2020 forum; 2020. p. 1859.
[157] Pulickan S, Lafon P, Langlois L. Stochastic geometric model for overlapping beads fabricated using wire arc additive manufacturing under uncertainties. Int J Adv Manuf Technol Jan. 2026;142:4103–18. https://doi.org/10.1007/s00170-026-17387-0.
[158] Lee J, Jadhav S, Kim DB, Ko K. Preliminary results for a data-driven uncertainty quantification framework in wire+arc additive manufacturing using bead-on-plate studies. Int J Adv Manuf Technol Apr. 2023;125(11–12):5519–40. https://doi.org/10.1007/s00170-023-11015-x.