Professional Certificate in Artificial Intelligence for Welding Processes

Reinforcement Learning

Action – the decision taken by the learning agent at a particular time step, influencing the environment's state. Related terms: Action Space, Policy, State. Explanation: In reinforcement learning (RL) for welding processes, an action could be selecting a welding current, adjusting travel speed, or changing shielding gas flow. The agent observes the current welding state (e.G., Temperature, bead geometry) and chooses an action that aims to improve weld quality. Example: If the sensor reports excessive heat input, the agent may reduce the current for the next segment. Practical application: Real-time adaptive welding controllers use actions to maintain consistent penetration depth across varying joint gaps. Challenges: Defining a discretized set of actions that captures the necessary granularity without exploding computational complexity; handling continuous-action spaces with precision.

Action Space – the complete set of actions that an RL agent can take in a given environment. Related terms: Discrete Action Space, Continuous Action Space, Action. Explanation: For welding, the action space may be discrete (e.G., Low/medium/high current) or continuous (any current value within a range). The choice impacts algorithm selection; Q-learning works well with discrete spaces, while policy-gradient methods handle continuous spaces. Example: A continuous action space could be a current range of 100–250 A with a resolution of 0.1 A. Practical application: Adaptive welding robots that smoothly vary voltage to compensate for heat accumulation. Challenges: Large or high-dimensional action spaces increase exploration difficulty and may require function approximation via neural networks.

Advantage Function – a measure of how much better a particular action is compared to the average action at a given state. Related terms: Value Function, Q-Function, Policy Gradient. Explanation: The advantage $A(s,a) = Q(s,a) – V(s)$ helps reduce variance in policy-gradient updates. In welding RL, it highlights actions that improve bead quality relative to the baseline policy. Example: If the baseline policy yields a bead width of 6 mm and a specific action reduces it to 5 mm, the advantage is positive, encouraging the agent to repeat that action. Practical application: Fine-tuning welding parameters in a stochastic environment where sensor noise may obscure raw rewards. Challenges: Accurate estimation of the advantage requires reliable value function approximators; bias can mislead the learning process.

Agent – the autonomous entity that interacts with the welding environment, perceives states, selects actions, and learns from rewards. Related terms: Environment, Policy, Reward. Explanation: In the Professional Certificate context, the agent may be a software module embedded in a welding power source, using RL to optimise process parameters. Example: An agent receives temperature sensor data, decides on current adjustments, and receives a reward based on bead uniformity. Practical application: Deploying agents on industrial CNC welding stations to achieve consistent quality across production batches. Challenges: Ensuring the agent respects safety constraints, such as maximum allowable current, while still exploring to discover optimal strategies.

Algorithm – the computational procedure that updates the agent's policy or value estimates based on

observed data. Related terms: Q-Learning, Deep Q-Network (DQN), Proximal Policy Optimization (PPO). Explanation: Different RL algorithms suit different welding scenarios. Model-free algorithms like DQN learn directly from interaction, while model-based approaches predict future weld outcomes. Example: Using PPO to train a welding robot in simulation before transferring the policy to the real machine. Practical application: Rapid prototyping of welding strategies with simulated arc physics, reducing wear on physical equipment. Challenges: Balancing sample efficiency (few interactions) with stability; avoiding catastrophic forgetting when the welding environment changes.

Artificial Neural Network (ANN) – a parametric function approximator composed of interconnected layers of neurons, used to model complex relationships. Related terms: Deep Learning, Function Approximation, Policy Network. Explanation: In RL for welding, ANNs often serve as policy or value networks, mapping sensor inputs to actions or expected returns. Convolutional layers may process visual data (e.G., Infrared images of the weld pool). Example: A feed-forward ANN takes arc voltage, current, and wire feed speed as inputs and outputs a probability distribution over possible current adjustments. Practical application: End-to-end learning where raw sensor streams are directly transformed into control commands without manual feature engineering. Challenges: Overfitting to limited welding data; ensuring the network remains interpretable for certification and safety audits.

Bellman Equation – a recursive relationship that defines the value of a state as the immediate reward plus the discounted value of the next state. Related terms: Dynamic Programming, Value Function, Temporal Difference (TD) Learning. Explanation: The Bellman equation underpins many RL algorithms. For welding, the state could be the current temperature profile, and the reward might reflect bead geometry quality. Example: $V(s) = r(s,a) + \gamma \cdot E[V(s')]$, where γ is the discount factor and s' is the state after taking action a. Practical application: Computing optimal welding trajectories by solving the Bellman optimality equation in a discretised state-action grid. Challenges: Exact solutions are intractable for high-dimensional welding states; approximations introduce bias that must be managed.

Batch Learning – training the RL model using a fixed dataset collected beforehand, rather than online interaction. Related terms: Offline RL, Replay Buffer, Experience Replay. Explanation: In welding contexts where live experimentation is costly, batch learning enables the agent to learn from historical process logs. Example: Using a dataset of 10,000 welds with associated sensor readings and quality outcomes to train a DQN offline. Practical application: Retrofitting legacy welding equipment with RL-based optimization without disrupting production lines. Challenges: Distribution shift between the logged data and the environment the agent will later encounter; ensuring sufficient coverage of state-action space in the batch.

Behavior Cloning – a supervised learning technique where the agent imitates expert demonstrations rather than learning from rewards. Related terms: Imitation Learning, Expert Policy, Dataset Aggregation (DAgger). Explanation: For welding, experienced operators can provide demonstration trajectories (e.G., Current profiles) that the agent copies before fine-tuning with RL. Example: Recording the current waveform of a skilled welder and training a neural network to predict the same waveform given sensor inputs. Practical application: Accelerating RL convergence by initializing the policy with human-derived good practices. Challenges: Demonstrations may not cover edge cases; the cloned policy may inherit suboptimal habits that RL must later correct.

Discount Factor (γ) – a scalar between 0 and 1 that determines how future rewards are weighted relative to immediate rewards. Related terms: Reward Horizon, Temporal Discounting, Value Function. Explanation: In welding, a higher γ places more emphasis on long-term weld quality (e.G., Minimizing residual stress), while a lower γ focuses on immediate bead appearance. Example: Setting γ = 0.99 To value the cumulative effect of a series of actions over an entire joint. Practical application: Designing reward structures that balance short-term defect avoidance with long-term structural integrity. Challenges: Choosing γ that reflects the true engineering trade-offs; overly high γ can cause instability in learning.

Exploration – the process by which an RL agent tries actions that it has not previously taken to discover potentially better strategies. Related terms: Exploitation, Epsilon-Greedy, Intrinsic Motivation. Explanation: For welding, exploration may involve testing unconventional current-feed speed combinations to uncover new optimal settings. Example: An ε-greedy policy selects a random action 5 % of the time, allowing the agent to sample rarely used parameter sets. Practical application: Periodic exploration phases during low-production periods to safely gather new data. Challenges: Ensuring exploration does not violate safety limits or produce defective welds; managing the trade-off between learning speed and production quality.

Experience Replay – a memory buffer that stores past transitions (state, action, reward, next state) for random sampling during training. Related terms: Replay Buffer, Off-Policy Learning, Batch Learning. Explanation: In welding RL, experience replay stabilises learning by breaking correlations between consecutive samples and reusing rare high-quality transitions. Example: Storing 50,000 weld episodes and randomly sampling mini-batches to update a DQN. Practical application: Leveraging scarce high-quality weld data multiple times to improve policy robustness. Challenges: Managing buffer size on embedded controllers; prioritising rare but informative experiences without biasing the learning process.

Exploit-Explore Trade-off – the balance between using known good actions (exploitation) and trying new actions (exploration). Related terms: Exploration, Exploitation, Policy Optimization. Explanation: In welding, over-exploitation may lock the system into suboptimal parameter sets, while excessive exploration can cause unacceptable defects. Example: Adaptive ε-greedy schedules that reduce ε as the agent's performance stabilises. Practical application: Dynamic adjustment of exploration rates based on real-time quality metrics, such as bead width variance. Challenges: Detecting when the learning plateau is due to true optimality versus insufficient exploration.

Function Approximation – the use of parametric models (e.G., Neural networks) to estimate value functions or policies when the state-action space is too large for tabular methods. Related terms: Neural Network, Linear Approximation, Generalisation. Explanation: Welding processes produce high-dimensional sensor streams (temperature, voltage, visual images). Function approximators map these to expected returns. Example: A convolutional neural network processes infrared weld pool images to predict the Q-value for each possible current adjustment. Practical application: Real-time prediction of weld defects, enabling the agent to intervene before a flaw propagates. Challenges: Ensuring the approximator does not extrapolate wildly outside the training distribution; maintaining computational efficiency on edge devices.

Generalisation – the ability of a trained RL model to perform well on unseen welding scenarios, such as different material thicknesses or joint geometries. Related terms: Overfitting, Domain Adaptation, Transfer Learning. Explanation: A policy that works for 2 mm stainless steel may need to generalise to 5 mm carbon

steel. Techniques like data augmentation and regularisation help achieve this. Example: Training on a mixture of simulated and real weld data to improve robustness across varying conditions. Practical application: Deploying a single RL controller across multiple welding cells with differing specifications. Challenges: Capturing the full variability of real-world welding environments in the training set; avoiding catastrophic performance drops when encountering out-of-distribution states.

Greedy Policy – a deterministic policy that always selects the action with the highest estimated value. Related terms: Epsilon-Greedy, Exploitation, Policy Improvement. Explanation: After sufficient learning, a welding agent may adopt a greedy policy to maximise quality, selecting the current-feed speed pair with the highest predicted reward. Example: Choosing the action a* = argmax_a Q(s,a) for the current sensor state s. Practical application: Production mode where the agent operates with minimal exploration to ensure consistent welds. Challenges: A purely greedy policy cannot adapt to sudden changes (e.G., A new material) without re-learning.

Hierarchical Reinforcement Learning (HRL) – an approach that decomposes a complex task into subtasks, each with its own policy, managed by a higher-level controller. Related terms: Options Framework, Sub-policy, Temporal Abstraction. Explanation: Welding a multi-pass joint can be split into “pass-1”, “pass-2”, etc., With each sub-policy handling local parameter tuning while the high-level policy decides when to switch passes. Example: An option representing “maintain constant heat input” is invoked during the filler-pass, while another option controls “ramp-down” for the final pass. Practical application: Reducing learning complexity for multi-stage welding processes, enabling faster convergence. Challenges: Designing appropriate subtask boundaries; ensuring seamless coordination between levels to avoid conflicts.

Importance Sampling – a statistical technique that re-weights samples from one distribution to estimate expectations under another distribution. Related terms: Off-Policy Evaluation, Weighted Importance Sampling, Policy Gradient. Explanation: When evaluating a new welding policy using data collected from an older policy, importance sampling corrects for the distribution mismatch. Example: Computing the expected reward of a new current-control policy using historic logs collected under a constant-current baseline. Practical application: Safe offline testing of candidate policies before deployment on live equipment. Challenges: High variance when the target and behavior policies differ significantly; requires careful clipping or truncation strategies.

Intrinsic Motivation – internally generated rewards that encourage the agent to explore novel states, independent of external task rewards. Related terms: Curiosity-Driven Exploration, Novelty Bonus, Exploration. Explanation: In welding, intrinsic motivation can drive the agent to try rarely used parameter combinations, revealing hidden performance regimes. Example: Adding a bonus proportional to the prediction error of a forward model of the weld pool temperature. Practical application: Accelerating discovery of optimal welding windows for new alloy families. Challenges: Balancing intrinsic and extrinsic rewards to avoid reckless exploration that harms product quality.

Markov Decision Process (MDP) – the formal framework for RL, consisting of states, actions, transition probabilities, rewards, and a discount factor. Related terms: State, Action Space, Policy. Explanation: Welding control can be modelled as an MDP where each state captures sensor readings, each action adjusts process parameters, and the reward reflects weld quality metrics. Example: S = {arc voltage, temperature gradient,

visual defect indicator}; A = {increase current, decrease speed, hold}. Practical application: Providing a mathematically rigorous basis for designing optimal welding controllers. Challenges: Accurately estimating transition dynamics in the presence of stochastic disturbances (e.G., Spatter, shielding gas fluctuations).

Model-Based RL – techniques that learn or exploit a model of the environment's dynamics to plan or generate synthetic experience. Related terms: Dynamics Model, Planning, Model-Free RL. Explanation: A welding agent may learn a predictive model of how current changes affect the temperature field, then use model-predictive control (MPC) to select actions. Example: Training a neural network to predict next-step temperature distribution given current, voltage, and feed speed. Practical application: Reducing real-world interactions by simulating many roll-outs of the welding process before updating the policy. Challenges: Model bias can mislead planning; high-fidelity welding physics are computationally intensive to model accurately.

Monte Carlo Methods – approaches that estimate value functions by averaging returns from complete episodes, without bootstrapping. Related terms: Episode, Return, Temporal Difference. Explanation: For batch welding tasks (e.G., A single joint), Monte Carlo estimates can be used to assess the long-term impact of a parameter sequence on residual stress. Example: Running a full weld simulation, recording the total reward, and averaging over many random parameter sequences. Practical application: Evaluating the expected quality of a welding schedule before committing to production. Challenges: Requires many complete episodes; not suitable for continuous online control where episodes are long or indefinite.

Neural Architecture Search (NAS) – automated methods for discovering optimal network structures for a given task. Related terms: Hyperparameter Optimization, Deep RL, Function Approximation. Explanation: In welding RL, NAS can identify the most effective network depth, width, and activation functions for processing sensor streams. Example: Using reinforcement-learning-based NAS to evolve a convolution-LSTM architecture that predicts weld pool dynamics. Practical application: Tailoring compact models for edge deployment on welding power sources with limited memory. Challenges: Computational cost of search; risk of over-fitting to the training dataset.

Off-Policy Learning – learning a target policy using data generated by a different behaviour policy. Related terms: Experience Replay, Importance Sampling, Q-Learning. Explanation: In welding, an operator may run a safe baseline process while the RL agent learns from those trajectories, updating a more aggressive policy offline. Example: Using DQN to learn a policy that maximises welding speed while the logged data comes from a conservative manual setting. Practical application: Incremental improvement of welding productivity without interrupting existing production schedules. Challenges: Distribution mismatch; ensuring the learned policy does not exploit artefacts present only in the historical data.

On-Policy Learning – learning while following the same policy that is being updated. Related terms: Policy Gradient, Actor-Critic, Exploration. Explanation: Algorithms like PPO require the agent to collect fresh data using the current policy, ensuring updates reflect the true interaction dynamics. Example: An RL welding controller periodically pauses production to collect a short batch of data using the latest policy before updating it. Practical application: Continuous improvement loops where the controller adapts in near-real time to subtle changes in material composition. Challenges: Balancing the need for fresh data with production throughput; managing safety during exploratory phases.

Policy – the mapping from states to actions that the RL agent follows. It can be deterministic or stochastic. Related terms: Stochastic Policy, Deterministic Policy, Policy Gradient. Explanation: In welding, a stochastic policy may output a probability distribution over current adjustments, allowing exploration, while a deterministic policy directly selects the best-estimated current. Example: Π(a|s) = Softmax(Q(s,·)) for a stochastic policy; a = μ(s) for a deterministic policy. Practical application: Deploying stochastic policies during low-risk training phases, then switching to deterministic policies for production. Challenges: Ensuring the stochastic component does not cause unacceptable variability in weld quality; calibrating the temperature of the softmax distribution.

Policy Gradient – a family of methods that directly optimise the parameters of a stochastic policy by gradient ascent on expected reward. Related terms: REINFORCE, Actor-Critic, Proximal Policy Optimization (PPO). Explanation: Policy-gradient methods are well-suited for continuous-action welding problems because they can output precise parameter values. Example: Updating policy parameters θ using $\nabla_\theta J \approx \mathbb{E}[\nabla_\theta \log \pi_\theta(a|s) \cdot A(s,a)]$, where A is the advantage. Practical application: Learning smooth current trajectories that minimise spatter while preserving penetration depth. Challenges: High variance of gradient estimates; requiring baseline subtraction or variance-reduction techniques to stabilise learning.

Q-Function (Action-Value Function) – the expected return of taking a particular action in a given state and thereafter following a specific policy. Related terms: Bellman Equation, Deep Q-Network (DQN), Value Function. Explanation: In welding RL, Q(s,a) predicts the quality reward of applying a certain current for the next time step, given the current sensor state. Example: $Q(s,a) = r(s,a) + \gamma \cdot \max_{a'} Q(s',a')$ for the optimal Q-learning update. Practical application: Using Q-values to rank candidate welding parameter sets in real time. Challenges: Approximating Q accurately in continuous state spaces; avoiding overestimation bias in DQN variants.

Replay Buffer Size – the capacity of the experience replay memory, typically measured in number of transitions. Related terms: Experience Replay, Batch Learning, Prioritized Replay. Explanation: A larger buffer stores more diverse welding experiences, improving learning stability, but consumes more memory on embedded controllers. Example: Setting the buffer to 100,000 transitions for a high-throughput welding robot. Practical application: Maintaining a rolling window of recent welds to capture drift in sensor calibration. Challenges: Determining an appropriate size that balances representation diversity with hardware constraints.

Reward Shaping – the process of designing additional reward components to guide learning toward desired behaviours. Related terms: Reward Function, Sparse Reward, Potential-Based Shaping. Explanation: Raw weld quality metrics (e.G., Defect count) may be sparse; shaping adds intermediate rewards such as “maintain temperature within target band”. Example: $R = -|\text{penetration\_error}| - \lambda \cdot |\text{temperature\_deviation}|$, where λ balances the two terms. Practical application: Encouraging the agent to keep the weld pool surface smooth during the entire pass, not just at the end point. Challenges: Avoiding unintended incentives that lead to gaming the reward (e.g., oscillating parameters to maximise intermediate bonuses).

Sample Efficiency – a measure of how many environment interactions are needed for the agent to achieve a certain performance level. Related terms: Data Efficiency, Off-Policy Learning, Model-Based RL. Explanation: Welding processes are expensive to run; high sample efficiency reduces the number of physical welds

required for training. Example: Achieving 90 % of optimal bead uniformity after only 500 real-world episodes. Practical application: Deploying RL on production lines with limited downtime for data collection. Challenges: Balancing sample efficiency with stability; integrating simulation data to boost efficiency while managing the reality gap.

Temporal Difference (TD) Learning – a class of methods that update value estimates based on the difference between successive predictions. Related terms: TD Error, TD(λ), Q-Learning. Explanation: TD learning combines Monte Carlo returns with bootstrapping, enabling online updates after each welding step. Example: $\Delta = r + \gamma \cdot V(s') - V(s)$ is the TD error used to adjust V(s). Practical application: Real-time adjustment of welding parameters after each micro-second of arc exposure. Challenges: Choosing λ to balance bias and variance; ensuring TD errors remain bounded in noisy sensor environments.

Transfer Learning – reusing a model trained on one task or domain as a starting point for a related task. Related terms: Domain Adaptation, Fine-Tuning, Pre-training. Explanation: A policy learned on mild-steel welding can be fine-tuned for high-strength steel with fewer additional samples. Example: Initialising a DQN with weights from a simulation of aluminum welding, then continuing training on copper. Practical application: Rapid deployment of RL controllers across multiple welding lines with differing material specifications. Challenges: Negative transfer where prior knowledge hinders learning; detecting when a source model is appropriate.

Value Function (State-Value Function) – the expected return when starting from a given state and following a particular policy thereafter. Related terms: Bellman Equation, Policy Evaluation, Advantage Function. Explanation: V(s) estimates the long-term weld quality achievable from the current sensor configuration. Example: $V(s) = E[\Sigma_{t=0}^{\infty} \gamma^{t} r_{t} \mid s_0 = s]$. Practical application: Using V(s) as a baseline to compute advantage, reducing variance in policy-gradient updates. Challenges: Accurate estimation requires sufficient coverage of the state space; function approximation may introduce bias.

Variational Autoencoder (VAE) – a generative model that learns a latent representation of input data, useful for dimensionality reduction. Related terms: Latent Space, Generative Modeling, Representation Learning. Explanation: In welding RL, a VAE can compress high-resolution infrared images into a low-dimensional latent vector fed to the policy network. Example: Training a VAE on 10,000 weld pool images to obtain a 32-dimensional latent code. Practical application: Enabling the RL agent to process visual data on low-power edge devices. Challenges: Ensuring the latent space retains task-relevant features; avoiding mode collapse.

Weighted Importance Sampling (WIS) – a technique that normalises importance-sampling weights to reduce variance. Related terms: Importance Sampling, Off-Policy Evaluation, Bias-Variance Trade-off. Explanation: When evaluating a new welding policy using historical data, WIS scales each sample's contribution by its weight divided by the sum of all weights. Example: $W_i = \pi_{new}(a_i|s_i) / \pi_{old}(a_i|s_i)$; normalized weight = $w_i / \Sigma_j w_j$. Practical application: Safer offline policy testing where raw importance sampling would otherwise produce high-variance estimates. Challenges: Still sensitive to extreme weight ratios; may require clipping strategies.

Zero-Shot Generalisation – the ability of a model to perform correctly on a task it has never seen during

training. Related terms: Transfer Learning, Meta-Learning, Domain Generalisation. Explanation: An RL welding controller that can immediately adapt to a new alloy without additional data exemplifies zero-shot capability. Example: Deploying a policy trained on steel and aluminium to a titanium welding cell, relying on shared latent representations. Practical application: Rapid rollout of RL controllers across newly introduced welding processes. Challenges: Requires highly abstract representations; risk of severe performance degradation if the unseen domain deviates too far.

Eligibility Trace – a temporary record of visited states and actions used to assign credit over multiple time steps. Related terms: TD(λ), Temporal Difference, Credit Assignment. Explanation: In welding, an eligibility trace can propagate the effect of a current adjustment to later defects observed several milliseconds later. Example: $E_t = \gamma \cdot \lambda \cdot e_{t-1} + \nabla\theta \log \pi\theta(a_t|s_t)$. Practical application: Enabling faster learning of delayed effects such as residual stress accumulation. Challenges: Choosing λ to balance short-term and long-term credit; managing trace memory on constrained hardware.

Exploration Noise – stochastic perturbations added to deterministic actions to encourage exploration. Related terms: Ornstein-Uhlenbeck Process, Gaussian Noise, Exploration. Explanation: For continuous welding control, adding noise to the current command helps the agent discover smoother trajectories. Example: $A_t = \mu(s_t) + \varepsilon_t$, where $\varepsilon_t \sim N(0, \sigma^2)$. Practical application: Incremental refinement of welding speed profiles during low-risk training runs. Challenges: Setting noise magnitude to avoid violating safety limits; decaying noise appropriately as learning progresses.