4th Holistic Innovation in Additive Manufacturing (HI-AM) Conference, Waterloo, Canada, 2021.
https://nserc-hi-am.ca/2021/

Sensor-Based In-situ Process Control of Robotic Wire Arc Additive Manufacturing Integrated with Reinforcement Learning

Yeon Kyu Kwak[1,2], Thomas Lehmann[1], Mahdi Tavakoli[2], Ahmed Qureshi[1]

1- Additive Design and Manufacturing Systems (ADaMS) Lab, Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta, T6G 2R3, Canada

2- Telerobotic and Biorobotic Systems (TBS) group, Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, T6G 2R3, Canada
Email Address: ykwak@ualberta.ca; lehmann@ualberta.ca; ajqureshi@ualberta.ca; mahdi.tavakoli@ualberta.ca

Abstract: *Wire and Arc Additive Manufacturing (WAAM) is a manufacturing technique capable of fabricating large-scale metallic components in a layer-by-layer fashion. As an emerging technology, there still exist numerous challenges that need to be overcome to ensure the geometrical accuracy of the part produced. With an increasing number of deposited layers, geometrical errors often accumulate in height and the accumulated heat becomes significant, leading to the slumping of the beads. The quality of the part can be enhanced through in-situ real-time feedback control. However, as the WAAM process is a time-variant process that is highly non-linear and multi-dimensional, it is difficult to model the process relating the process parameters to the final quality of the produced part. To address this challenge, a sensor-based in-situ process control framework integrated with reinforcement learning (RL) artificial intelligence (AI) is proposed to iteratively learn the impacts of various process parameters to finally control the output geometry of a single-bead multi-layer part. The proposed control frameworks are then implemented and simulated on a robotic large-scale WAAM system.*

Keywords: *Additive manufacturing, Process control, Reinforcement learning*

1. Introduction

Interest in wire arc additive manufacturing (WAAM) has grown significantly in recent years due to its numerous advantages over traditional subtractive manufacturing [1]. It is capable of fabricating large-scale complex metallic components as well as have a reduced buy-to-fly ratio [2]. One of the main challenges that limit the full potential of WAAM is its lack of manufacturing accuracy. As WAAM fabricates a component in a layer-by-layer fashion, a buildup of an error may occur where a small error in a previous layer would gradually build up throughout every layer, further negatively affecting the geometrical accuracy of produced part. There exist various input parameters that affect the geometrical accuracy of the final part, and they are often difficult to control as they are highly non-linear and coupled [3]. To overcome this challenge, control of process parameters is required as it would be able to rectify errors and correct itself throughout the manufacturing process. However, WAAM is a very complex time-variant dynamic process with many different process parameters. Some of the input process parameters include torch positioning and speed, wire feed rate, voltage, and current. Observable process parameters include thermal, geometrical information of beads at the location of the deposition.

Heralic et al. [4] used a 3D laser scanner to obtain a profile of each layer after every deposition. Through iterative learning control, the deviation of height was adjusted by controlling the wire feed speed for the next deposition layer. Xiong et al. [5] established an improved self-learning neuron feedback control of bead width with a visual sensor and its corresponding image processing algorithm. Doumanidis and Kwak [6, 7] used a laser scanner and infrared sensor to monitor the gas metal arc welding (GMAW) system. A simultaneous but independent closed-loop control of bead width and reinforcement height to the desired specification was achieved. Smith et al. [8] used a CCD camera to capture the image of the molten pool surface and obtained the width of its molten pool. This data was then used on a closed-loop control of a GTAW system as a feedback signal to control weld penetration. Fan et al. [9] implemented feedback control to monitor welding penetration using temperature data. An infrared sensing system monitors the surrounding temperature of the melt pool during a welding process. Liu and Zhang [10, 11] developed a linear-model-based predictive controller to control the 3D weld pool geometry of a GTAW process. Dharmawan et al. [12] proposed a reinforcement learning control framework for controlling layer height. The wire feed rate and torch travel speed were dynamically adjusted according to the measured height using a laser 3D scanner.

Despite the effort of modeling and controlling the WAAM process, in the above reviewed literature, there are other relevant parameters that were not considered. For instance, one would optimize bead height but not width. Often one of the process parameters

such as the torch travel speed is held constant in a feedback control loop. This called for a use of control algorithms and techniques that takes into account many more of the process parameters.

This paper proposes a method of sensor-based in-situ control of robotic WAAM integrated with reinforcement learning (RL) techniques. The algorithm used is called Q-learning [13], also known as a model-free off-policy Temporal Difference method. The control algorithm is applied to the WAAM system to iteratively learn the set of values for each of the various process parameters to achieve a specified geometrical quality. After the algorithm converges with the Q-learning method, the system can effectively identify what set of action is best for the system to deploy in a real-time manner in a single-track, multi-layer printing scenario. The major advantage of the method is that it can adjust wire feed rate, torch standoff distance, torch travel speed, and voltage in accordance with real-time sensory information from a profilometer and an infrared camera to achieve specified geometrical quality. The work is performed on a simulation. It is preliminary and is a beginning step towards achieving a goal of further improving print quality using intelligent in-situ WAAM control.

2. Reinforcement learning – Q-learning

Reinforcement learning (RL) is a type of Machine learning (ML) approach that is used to train an entity called an *agent* to accomplish a specific task. Fig.1 depicts the interaction of agent and environment in RL. The agent is the entity that takes some *actions*. These actions may impact the time-variant environment and can be modeled as a Markov Decision Process (MDP) [14]. This environment is then observed and returned to the agent in a form of *state* and *reward*. The reward is a numerical value that represents the quality figure for the last action performed by the agent. This agent and environment interaction is illustrated in Fig.1. Through an iterative process, the agent is able to learn the optimal *policy*, which is a rule for the agent to select some action given a certain state.

Q-learning is one of the most well-known and employed RL algorithms that belong to the class of off-policy methods as convergence is guaranteed for any agent's policy [14]. The basis of Q-Learning stems from a concept of *Quality Matrix* or Q-Matrix. With a matrix size of $N \times Z$ where N is the number of possible actions and Z is the number of possible actions that can be taken by the agent. Thus, the state action space $S \times A$ is discrete. The Q-Matrix is populated with Q-values that represent "how good" is it to take specific action given the current state. Algorithm 1 summarizes the general Q-learning method.

The algorithm begins with initialized Q-matrix with a random value and is updated using the Bellman optimality equation (1).

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[R_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (1)$$

Variables in (1) are defined as,

- s_t and s_{t+1}: current and next state of the observed environment, where $s_t \in \mathbb{S}$ and S is the set of possible states.
- a_t and a_{t+1}: current and next action taken by the agent, where $a_t \in \mathbb{A}(\mathbb{S}_t)$ is the set of possible actions given state.
- γ: discount factor $\gamma \in [0, 1]$. Defines how much of future rewards are taken into account instead of the immediate rewards.
- α : learning reate, $\alpha \in [0, 1]$. Defines how much of newest knowledge has to replace the older one.
- R_t: numerical value of an immediate reward, a consequence of the action, a taken.

Algorithm 1. Q-learning method [13]

Set algorithm parameters: α, γ
Initialize the Q-matrix, $Q(s, a)$ for all $s \in \mathbb{S}$, $a \in \mathbb{A}$, arbitrarily
Repeat for every episode:
  Initialize s
  Loop for each step of episode:
    Choose a_t from s_t with a set policy derived from Q (use ϵ-greedy)
    Take action a_t and observe reward, R and next state s_{t+1}
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$ (update Q-matrix)
    $s_t \leftarrow s_{t+1}$
  until s_t is terminal

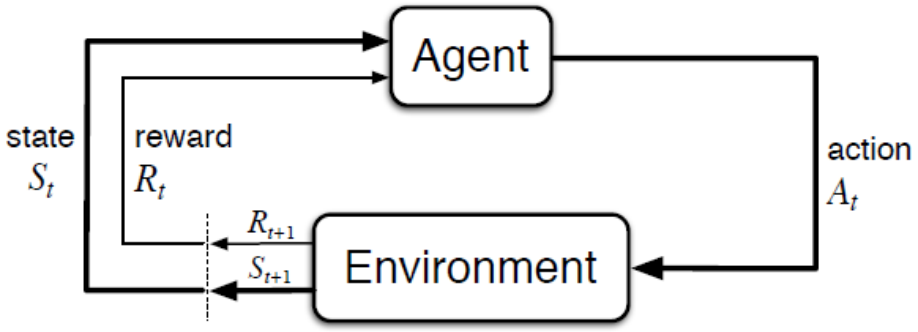

Fig. 1 Agent-environment interaction

The type of Q-learning deployed here is specifically a ϵ-greedy Q-learning. ϵ-greedy method is a simple probabilistic exploratory technique commonly used in RL. ϵ represents a value of range $[0, 1]$ at which if a randomly generated number between that range falls under, the agent takes a completely random action given a state. Otherwise, take a best-known action.

3. Implementation of Q-learning in WAAM

The conceptual idea of reinforcement learning is translated into implementation in WAAM of single-track multi-layer wall. Fig. 2 demonstrates the flow of the system with the incorporation of the RL algorithm. The state of the environment corresponds to the real-time observation data from a profilometer and an IR camera. The profilometer measures the width and height of the bead that the deposition occurs at. Also, the IR camera provides the temperature data at the point of the deposition. The thermal and geometrical data of the previous layer largely affects the geometry of the next layer. With the two data combined, the agent is to take a corresponding optimized action of changing wire feed speed, torch travel speed, and torch standoff distance to specific values that would ultimately give the desired geometry of the next layer and the next and so forth. As the Q-learning algorithm works with discretized values, Table 1 and 2 is tabulated to show the equispaced and discretized values of various states and actions considered in this study.

Table 1. State or observed process parameters discretized within a specified range

State	Range	Discretized into counts of
Bead width at deposition point [mm]	[6, 14]	4
Bead height at deposition point [mm]	[2, 4]	4
Temperature at deposition point [C°]	[200, 700]	5

Table 2. Action or process input parameters discretized within a specified range

Action	Range	Discretized into counts of
Wire feed speed, WFS [m/min]	[2, 3]	10
Torch standoff distance, SOD [mm]	[10, 13]	3
Torch travel speed, TTS [cm/min]	[25, 35]	4

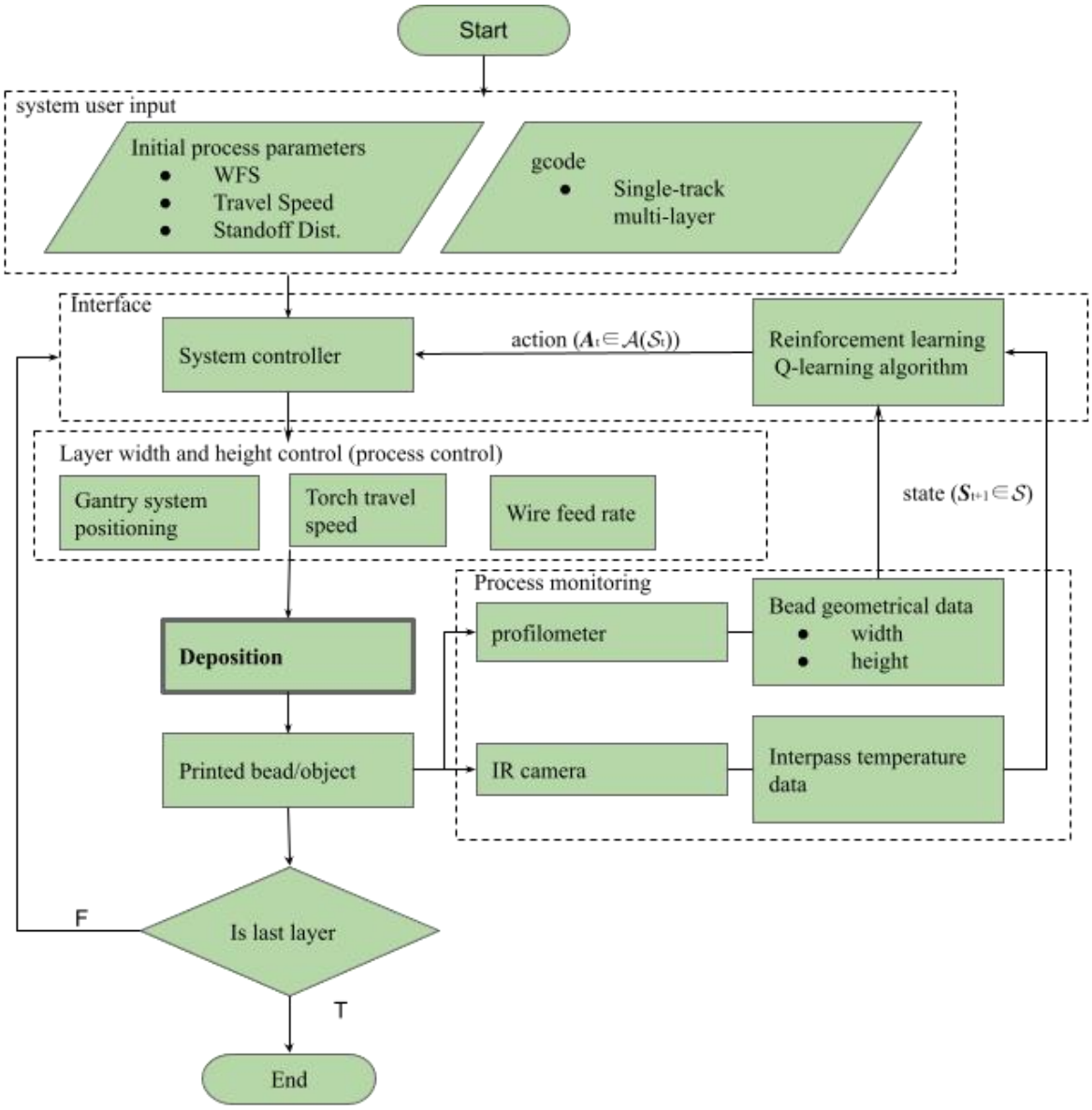

Fig. 2 System flow diagram for printing single-track multi-layer wall. Process monitoring gives data for RL to evaluate rewards and observe the state of the environment. Process control refers to the action taken by the agent.

The first layer is deposited with a commonly known process parameter and the profilometer mounted behind the torch records the bead profile. The setup can be seen in Fig. 4. Along with the known temperature and the geometrical profile of the previous layer, or a state, deposition of the next layer commences with specific values of wire feed speed, torch standoff distance, and torch travel speed, or an action. As the deposition of the next layer occurs, the profiler observes the geometrical data of the bead just deposited, given the state information of the previous bead's width, height, and temperature data. The geometrical data of the bead that just deposited is used to calculate the reward,

$$R_t = -|h_o - h_a| - |w_o - w_a| \qquad (2)$$

Where h_o and h_a represent the objective height and measured actual height, respectively. w_o and w_a represents the objective width and measured actual width, respectively. With the reward and through Bellman optimality equation (1), the Q-value can be obtained and be tabulated into the Q-matrix of Table 3. The Q-matrix is tabulated at every interval where the action parameter changes and a state observation occurs. The intervals at where action changes and state observation occurs is portrayed in Fig. 3. The process iterates for every episode where the terminal state is determined to be at the point where the $R_t < -0.5$.

Table 3. Q-matrix

State \ Action	$A_1 = (wfs_1, sod_1, tts_1)$	$A_2 = (wfs_1, sod_1, tts_2)$	...	$A_{75} = (wfs_5, sod_3, tts_5)$
$S_1 = (T_1, w_1, h_1)$	$Q(S_1,A_1)$	$Q(S_1,A_2)$	...	$Q(S_1,A_{75})$
$S_2 = (T_1, w_1, h_2)$	$Q(S_2,A_1)$	$Q(S_2,A_2)$		
...	...		...	
$S_{45} = (T_5, w_3, h_3)$	$Q(S_{45},A_1)$			$Q(S_{45},A_{75})$

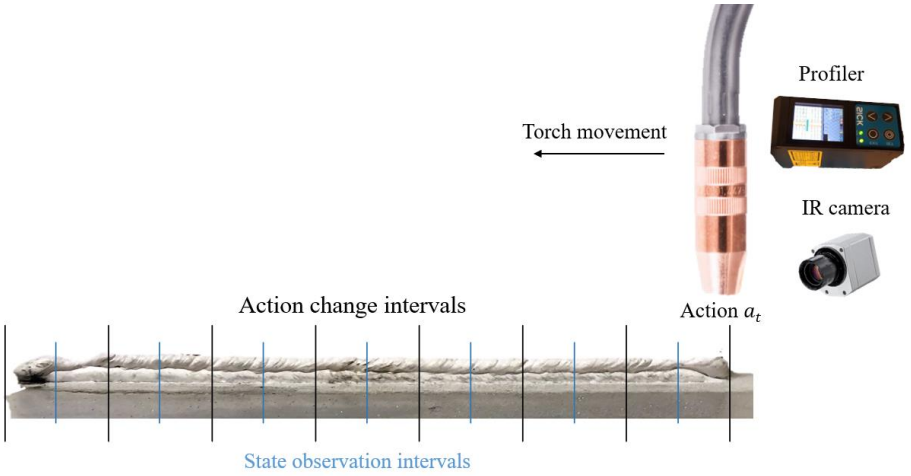

Fig. 3 Layout of the setup shows that the agents are discretized into sections. An action occurs and the result of the action is observed with the profiler and the IR camera. The observed profile data is used to evaluate the reward.

Fig. 4 Profilometer attached behind and along the trajectory of the print path

Being a preliminary work, the Q-learning method was validated using a simulator before commencing real-life experiments. The second-order regression model [15] was used to map the input parameters, namely the wire feed speed, standoff distance, torch travel speed, to the resulting width and height of the printed bead. The output temperature data was roughly simulated without an expert modeling equation. The learning rate α, discount factor γ, exploratory threshold ϵ, was set at 0.5, 0.99, and 0.1, respectively during the simulation.

4. Results and discussion

The simulation of the experiment was conducted to show the convergence. Fig. 6 shows that the first episode of learning had an average reward of approximately -1.9 which corresponds to the summed deviation of width and height from the desired value in units of mm. The individual result for deviation of width and height is shown in Fig. 7. This error is further minimized as the algorithm further tabulates the Q-Matrix. The system, over 300 episodes seem to converge at around a reward of -1.2.

Although cropped out for a visual understanding of the performance, in Fig.5 (right), it is notable that the first episode took an average of 300 iterations until reaching the terminal state and quickly down to 50 iterations for the next episode. This amount of iteration counts may or may not be a problem depending on how sparse the action change interval is in Fig. 3 in the real-life experiment. The steady error of the resulting graph is occurred from the value of ϵ, which is fixed throughout the entire simulation experiment. The major disadvantage of Q-learning is that it takes a long time and many iterations for the algorithm to reach the optimal Q-value. As the Q-learning learns a deterministic policy, the agent either chooses the best action or a random action. This could possibly be problematic in a non-stationary environment that is influenced by an unknown disturbance. As the states and actions are discretized, the resolution of state observed and the actions taken is limited to Table 1 and 2, respectively.

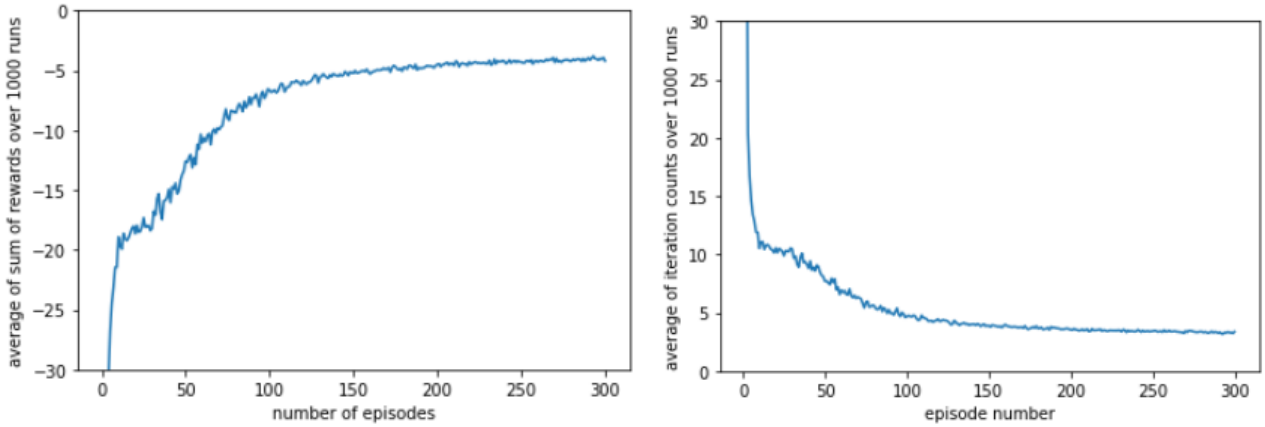

Fig. 5 Sum of reward during each episode (left), Number of actions taken by the agent until reaching the terminal state (right) averaged over 1000 independent runs

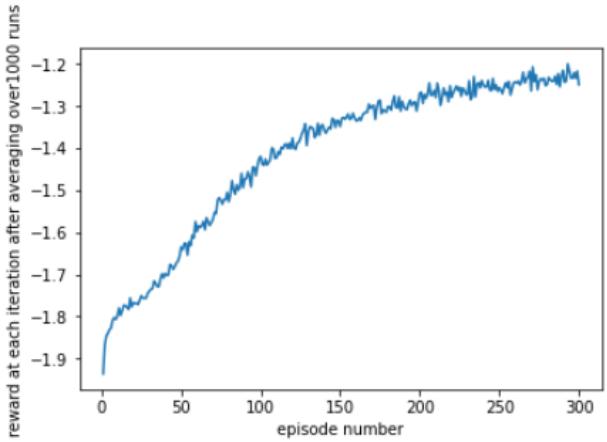

Fig. 6 Average reward observed per iteration, averaged over 1000 independent runs

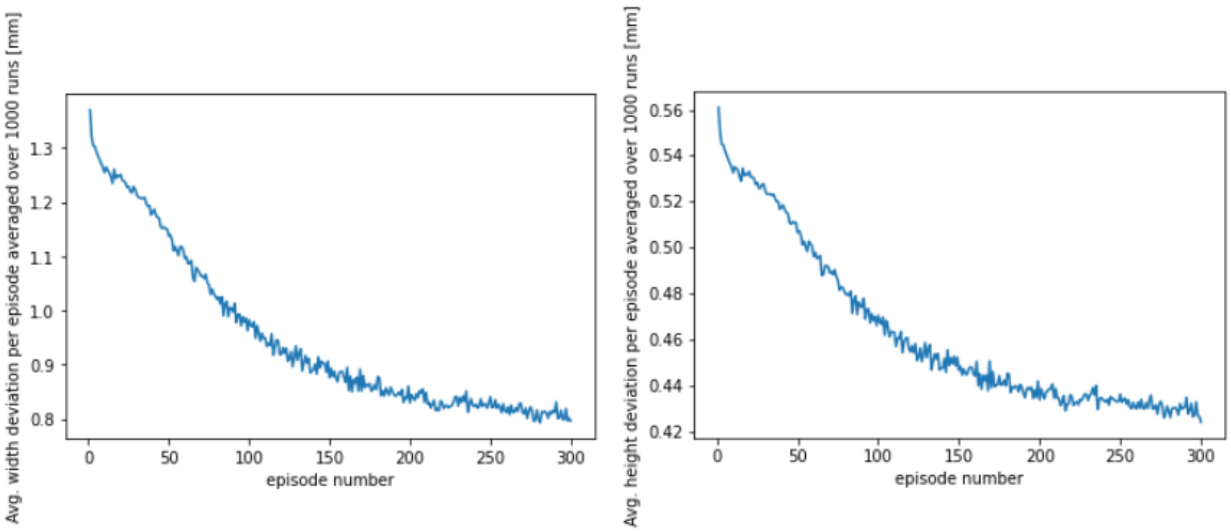

Fig. 7 Averaged absolute value of width and height deviations per episode averaged over 1000 independent runs

5. Conclusion and future work

The manuscript presents a preliminary study of sensor-based in-situ control of robotic wire arc additive manufacturing system integrated with a reinforcement learning technique called Q-learning. The reinforcement learning framework enabled the system to consider discretized values of wire feed speed, torch travel speed, and torch standoff distance as the system input while observing the bead geometry and temperature for closed-loop control. The result shows that the algorithm converges with a steady error of approximately 0.8 mm and 0.43 mm for width and height, respectively over iterations of many episodes. The encouraging preliminary result of the study opens more opportunities for improving WAAM systems in making the process more efficient and reliable.

The future work based on this outcome is to translate the simulation into real life. Also, the rate of convergence could be enhanced with lesser iteration and the steady error may be minimized by performing a hyperparameter study and deploying the decaying ϵ method. Other suitable RL algorithms that can take into account expert domain knowledge and model, and handle continuous sets of action and state parameters will be considered and explored.

Acknowledgment

This research is supported by the NSERC HI-AM Strategic Partnership Grant No. 494158.

References

[1] Xia, C., Pan, Z., Polden, J., Li, H., Xu, Y., Chen, S., & Zhang, Y. (2020). A review on wire arc additive manufacturing: Monitoring, control and a framework of automated system. *Journal of Manufacturing Systems*, *57*, 31-45.

[2] Williams, S. (2015). Wire+ arc additive manufacturing vs. traditional machining from solid: a cost comparison.

[3] Xia, C., Pan, Z., Polden, J., Li, H., Xu, Y., Chen, S., & Zhang, Y. (2020). A review on wire arc additive manufacturing: Monitoring, control and a framework of automated system. *Journal of Manufacturing Systems*, *57*, 31-45.

[4] Heralić, A., Christiansson, A. K., & Lennartson, B. (2012). Height control of laser metal-wire deposition based on iterative learning control and 3D scanning. *Optics and lasers in engineering*, *50*(9), 1230-1241.

[5] Xiong, J., Zhang, G., Qiu, Z., & Li, Y. (2013). Vision-sensing and bead width control of a single-bead multi-layer part: material and energy savings in GMAW-based rapid manufacturing. *Journal of cleaner production*, *41*, 82-88.

[6] Doumanidis, C., & Kwak, Y. M. (2001). Geometry modeling and control by infrared and laser sensing in thermal manufacturing with material deposition. *J. Manuf. Sci. Eng.*, *123*(1), 45-52.

[7] Doumanidis, C., & Kwak, Y. M. (2002). Multivariable adaptive control of the bead profile geometry in gas metal arc welding with thermal scanning. *International Journal of Pressure Vessels and Piping*, *79*(4), 251-262.

[8] Pires, J. N., Smith, J. S., & Balfour, C. (2005). Real-time top-face vision based control of weld pool size. *Industrial Robot: An International Journal*.

[9] Fan, H., Ravala, N. K., Wikle III, H. C., & Chin, B. A. (2003). Low-cost infrared sensing system for monitoring the welding process in the presence of plate inclination angle. *Journal of materials processing technology*, *140*(1-3), 668-675.

[10] Liu, Y. K., & Zhang, Y. M. (2013). Model-based predictive control of weld penetration in gas tungsten arc welding. *IEEE Transactions on Control Systems Technology*, *22*(3), 955-966.

[11] Liu, Y., & Zhang, Y. (2013). Control of 3D weld pool surface. *Control Engineering Practice*, *21*(11), 1469-1480.

[12] Dharmawan, A. G., Xiong, Y., Foong, S., & Soh, G. S. (2020, May). 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) (pp. 4030-4036). IEEE.

[13] Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8(3-4), 279-292.

[14] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press

[15] Xiong, J., Zhang, G., Hu, J., & Wu, L. (2014). Bead geometry prediction for robotic GMAW-based rapid manufacturing through a neural network and a second-order regression analysis. *Journal of Intelligent Manufacturing*, *25*(1), 157-163.