Guided Policy Search Based Control of a High Dimensional Advanced Manufacturing Process

Amit Surana and Kishore Reddy

Abstract—In this paper we apply guided policy search (GPS) based reinforcement learning framework for a high dimensional nonlinear optimal control problem arising in an additive manufacturing process. The problem comprises of controlling the process parameters so that layer-wise deposition of material leads to desired geometric characteristics of the resulting part surface while minimizing the material deposited. A realistic simulation model of the deposition process along with carefully selected set of guiding distributions generated based on iterative Linear Quadratic Regulator (LQR) is used to train a neural network policy using GPS. A closed loop control based on the trained policy and in-situ measurement of the deposition profile is tested experimentally, and shows promising performance.

I. INTRODUCTION

In this paper we apply guided policy search (GPS) a deep reinforcement learning (RL) approach for a high dimensional nonlinear optimal control problem in an advanced/additive manufacturing (AM) application. Deep Q-learning [1], [2], [3], and direct policy search [4] and actor-critic based deep RL (DRL) approaches [5], [6] have recently shown unprecedented success in dealing with high dimensional problems [7]. GPS [4] is a direct policy search method which uses a neural network (NN) to parameterize the policy, and transforms the policy search into a supervised learning problem, where the training set (which “guides” the policy search to regions of high reward) is generated by simple trajectory-centric algorithms. It employs a regularized importance sampled policy optimization to enable stable and efficient training [8]. We selected GPS for our application, as direct policy search methods scale gracefully with dimensionality and offer appealing convergence guarantees [9], [10]. Furthermore, compared to Q-learning or actor-critic based DRL methods, it is expected that GPS offers a better sample efficiency as it uses trajectory optimizer (e.g., iterative Linear Quadratic Regulator) to guide policy learning leading to a faster convergence towards optimum.

The problem considered in this paper comprises of controlling the process parameters during cold spray (CS), a metal AM process [11]. Specifically, the goal is to control the spray nozzle motion, so that layer-wise deposition of material leads to desired geometric characteristics of the resulting part surface while minimizing the material deposited. The relationship between the surface profile growth and nozzle motion is governed by a time dependent nonlinear partial differential equation (PDE). We discretize the PDE in space/time to obtain a finite dimensional nonlinear input-output system, and use a nonlinear optimization framework to calibrate the CS model parameters based on the experimental data. The NN policy is trained using GPS on the calibrated model using a carefully selected set of guiding distributions generated based on iterative Linear Quadratic Regulator (LQR). In order to ensure that the errors made by NN controller during layer deposition do not propagate to successive layers, we use suitable domain randomization. Closed loop control based on the trained NN policy was implemented in a lab setup with feedback from laser displacement sensor which measures the surface profile in real time during deposition. The experimental studies show a material saving of up to 35% compared to currently used open loop approaches for process parameter selection which rely on extensive offline experimental trial and error and/or expert knowledge, and do not account for in process uncertainties/disturbances.

Related work: The need for in-situ monitoring, machine learning and closed loop control has been recognized as a key enabler to improve process repeatability/reliability in AM applications [12], [13], [14]. Specifically, in the CS context, [15] apply a model predictive control (MPC) approach in simulations. We use a similar model/setup but employ a DRL framework, and provide an experimental demonstration. MPC can be computationally demanding for high dimensional nonlinear problems and thus pose real time deployment challenges. On the other hand, DRL can provide optimal solutions as MPC, but offers significant computational advantage as once policy has been learned offline, it can be executed online sufficiently fast [16].

II. COLD SPRAY CONTROL PROBLEM

CS is an AM process in which powder particles (typically 10 to 50 micron) are accelerated to very high velocities (200 to 1000 m/s) by a supersonic compressed gas jet at temperatures well below their melting point and then deposited on the surface to build new parts or repair structural damages such as cracks and unwanted indentations [11]. CS is a multi-scale process with complex physics [17], making it challenging to optimize the process parameters which result in desired part properties. Additionally, the nozzle motion relative to part/defect geometry needs to be programmed to achieve desired surface deposition profile. Currently, this is accomplished by manual trial and error per part, and often results in a conservative solution leading to excess material deposition which needs to be eventually machined out to meet the required geometric tolerances. Moreover, the nozzle motion once programmed remains fixed, and is not adapted...
to account for any process variations/disturbances during the operation. Such an approach of overbuilding could lead to a significant loss of expensive material.

The focus of our work is to develop a framework for feedback control of nozzle motion that minimizes the material wastage while achieving desired surface profile. While detailed CS models are highly complex, for our control development purposes it suffices to use a reduced order model described in [15] which captures the impact of nozzle motion/spray behavior on the dynamics of deposition process at a macroscopic scale. Let $p(t), h(t)$ and $\alpha(t)$ be the spray nozzle’s, position and orientation as a function of time, respectively, see Fig. 1a. Then the surface height profile $D(x,t) (x \in \mathbb{R})$ evolution is governed by a nonlinear integro-differential equation,

$$
\frac{\partial D(x,t)}{\partial t} = \int_0^T \phi(\tan(\beta(t)) - \tan(\alpha(t)))\psi \left( \frac{\tan(\beta(t)) - \frac{\partial}{\partial x} D(x,t)}{1 + \tan(\beta(t)) \frac{\partial}{\partial x} D(x,t)} \right) dt, \quad (1)
$$

where, $\tan(\beta(t)) = \frac{x-p(t)}{h(t)-D(x,t)}$ and the functions

$$
\phi(z) = 1 - \left(1 + \frac{\rho}{z} \right)^{-1}, \quad \psi(z) = \frac{1}{c} \left(0.5 + \frac{\tan(\pi \max(\left\lfloor z/b \right\rfloor,1))}{\pi} \right),
$$
capture the distribution of particles in the spray cone and the nozzle efficiency, respectively (see Fig. 1b). We will denote by $p = (\rho, a, b, c, \kappa)'$ as the model parameter vector, where $t$ denotes the vector transpose. In above we have restricted to a one-dimensional model, as often in applications the part is rotated while nozzle motion is confined along the axis of rotation, see Sec. IV.

The space-time discretization of Eqn. (1) leads to a discrete time nonlinear input-output system

$$
s_{t+1} = f(s_t, u_t; p), \quad (2)
$$

where, $s_t = (d_t', p_t, h_t, \alpha_t)'$ is the system state, $u_t = (v^h_t, v^\perp_t, \omega_t)'$ are the control inputs, and

$$
f(s_t, u_t) = \begin{pmatrix}
d_{t+1} + g_1(d_t, p_t, h_t, \alpha_t; p)dt \\
\vdots \\
d_{Ndt} + g_{Nd}(d_t, p_t, h_t, \alpha_t; p)dt \\
p_t + v^h_t dt \\
h_t + v^\perp_t dt \\
\alpha_t + \omega_t dt
\end{pmatrix}. \quad (3)
$$

Here, $d_t = (d_1 \cdots d_{Nd})'$ is the vector of discretized surface height profile $D(x,t)$, with $d_i$ being the surface height at the spatial location $x_i = x_0 + (i-1)dx, i = 1, \cdots, Nd$ and $Nd$ is the number of discretized cells of size $dx$, see Fig. 1a. We assume a kinematic model of the nozzle motion, so that the nozzle position/orientation can be controlled by changing its linear $v^h_t, v^\perp_t$ and angular $\omega_t$ speeds, respectively. $g_i$ represent the discretization of the integral term appearing in the Eqn. (1).

Given initial condition $s_0 = (d'_0, p_0, h_0, \alpha_0)'$, the objective is to determine control sequence $u_t = (v^h_t, v^\perp_t, \omega_t)', t = 1 \cdots, T$ such that the final surface profile matches a desired profile $d_f$,

$$
\min_{u_1, \cdots, u_T} \sum_{t=1}^{T-1} c(s_t, u_t) + c_e(s_T), \quad (4)
$$

where, $c_e(s_T) = ||d_T - d_f||^2$ is the terminal cost, and $c(s_t, u_t) = w_0 + w_1(v^h_t)^2 + w_2(v^\perp_t)^2 + w_3\omega_t^2$ is running cost which imposes penalties on nozzle’s speed and angular rate, and the total spraying time $T$ which is not known apriori and needs to be optimized as well. Additionally, bounds can be imposed on velocity and angular rates; $v^h_{\min} \leq v^h_t \leq v^h_{\max}, v^\perp_{\min} \leq v^\perp_t \leq v^\perp_{\max}$ and $\omega_{\min} \leq \omega_t \leq \omega_{\max}$, respectively. This optimal control problem can be solved using MPC, see [15]. However, since typically $Nd \gg 1$ to resolve the surface profile sufficiently, such an approach can be computationally demanding for real time deployment. We explore direct policy search based RL methods which are well suited for such high-dimensional control applications, since they scale gracefully with dimensionality and offer appealing convergence guarantees [9], [10]. However, it is often necessary to carefully choose a specialized policy class to learn the policy in a reasonable number of iterations without getting trapped into poor local optima.

III. GUIDED POLICY SEARCH

GPS is a recently proposed direct policy search method which uses a NN to parameterize the policy and transforms the policy search into a supervised learning problem, where the training set (which “guides” the policy search to regions of high reward) is generated by simple trajectory-centric algorithms. NN provide a general and flexible representation which can represent a broad range of behaviors. However, naive supervised learning will often fail to produce a good policy and a regularized importance sampled policy optimization has been proposed. We will use the end-to-end GPS formulation [4], which we briefly review. Consider a finite horizon stochastic optimal control problem

$$
\min_{\pi_\theta} \mathbb{E}_{\pi_\theta} \left[ \sum_{t=1}^{T} c(s_t, u_t) \right], \quad (5)
$$

where, $c(s, u)$ is the cost function, and the expectation $\mathbb{E}$ is taken under the randomized policy $\pi_\theta(u_t|s_t)$ which is parameterized by $\theta$ and the uncertain system dynamics with state transition probability $p(s_{t+1}|s_t, u_t)$. Let $q(u_t|s_t)$ be a guiding policy, then the problem (5) can be reformulated as an equivalent problem:

$$
\min_{q, \pi_\theta} \mathbb{E}_{q} \left[ \sum_{t=1}^{T} c(s_t, u_t) \right], q(u_t|s_t) = \pi_\theta(u_t|s_t), \forall t, s_t, u_t,
$$

which has an infinite number of constraints. To make the problem tractable one can use the moment matching ap-
proach, e.g.

$$\min_{q, \pi_\theta} \mathbb{E}_q \left[ \sum_{t=1}^{T} c(s_t, u_t) \right],$$

$$\mathbb{E}_{q(u_i|s_i)q(s_i)}[u_i] = \mathbb{E}_{\pi_\theta(u_i|s_i)q(s_i)}[u_i], \ \forall t. (6)$$

This constrained problem can be solved by a dual descent method, e.g. Bregman Alternative Direction Method of Multipliers (BADMM) Lagrangian formulation leads to

$$\mathcal{L}(q, \pi_\theta) = \sum_{t=1}^{T} \left[ \mathbb{E}_{q(s_t,u_t)}(c(s_t,u_t)) + \lambda t D_K(q, \pi_\theta) \right],$$

where we have taken the Bregmann divergence to be the KL divergence $D_KL(\pi_\theta|q)$. This leads to following iterative optimization scheme,

$$q \leftarrow \arg \min_q \sum_{t=1}^{T} \left[ \mathbb{E}_{q(s_t,u_t)}(c(s_t,u_t) - \lambda t u_t + \nu t \phi_t(q, \theta)) \right],$$

$$\theta \leftarrow \arg \min_{\theta} \sum_{t=1}^{T} \left[ \mathbb{E}_{\pi_\theta{}(u_t|s_t)q(s_t)}[\lambda t u_t + \nu t \phi_t(q, \theta)] \right],$$

$$\lambda_t \leftarrow \lambda_{t-1} + \alpha_t \left( \mathbb{E}_{\pi_\theta{}(u_t|s_t)q(s_t)}[u_t] - \mathbb{E}_{q(s_t,u_t)}[u_t] \right)$$

for $t = 1, \cdots, T$. Note that for each $t = 1, \cdots, T$, $\lambda_t$ is a vector of Lagrangian multipliers with same dimension as the control $u_t$. $\alpha$ can be chosen in range $[0, 1]$, lower values lead to better numerical stability. The weights $\nu_t$ are initialized to low values such as $0.01$ and incremented based on a schedule which adjusts the KL-divergence penalty to keep the policy and trajectory in agreement by roughly the same amount at all time steps. The above steps can be simplified under Gaussian assumptions:

- $\pi_\theta(u_t|s_t) \sim N\left(u_t; \mu^\pi(s_t; w), \Sigma^\pi\right)$, where the policy parameters are $\theta = (w, \Sigma^\pi)$. We will use a neural network (NN) with weights $w$ to represent the mean policy $\mu^\pi(s_t; w)$, while $\Sigma^\pi$ is policy covariance which is assumed to be state independent,

- $q = \sum_{t=1}^{M} q_t$ is mixture of Gaussians with $q_t(u_t|s_t) \sim N(u_t; \mu^\pi_t(s_t), \Sigma^\pi_t)$

where, $N(\cdot; \mu, \Sigma)$ is a multinomial normal probability distribution with mean vector $\mu$ and covariance matrix $\Sigma$. With this assumption, the overall GPS algorithm is summarized below. Steps 1-4 are repeated for a preselected $K$ iterations or else if a prescribed convergence criterion is met.

**Step 1:** Solve optimization (7) which under the Gaussian assumption simplifies to

$$\arg \min_{q_t} \sum_{t=1}^{T} \mathbb{E}_{q_t(s_t,u_t)} \left[ c_t(s_t,u_t) - \mathcal{H}(q_t(u_t|s_t)) \right],$$

where, $c_t(s_t, u_t) = \frac{c(s_t, u_t)}{p_v} - \frac{1}{p_v} u_t^T \lambda_t^\pi - \log(\pi_\theta(u_t|s_t))$ and, $\mathcal{H}$ is the standard differential entropy. A local optimal solution (which determines $\mu^\pi_t, \Sigma^\pi_t$) of the above problem around a selected nominal trajectory can be obtained via a variation of iterative LQR (iLQR) [18]. The nominal trajectory can be constructed in a variety of ways. e.g.
from demonstrations, using randomized control inputs or via solution obtained using MPC [19].

**Step 2:** Sample trajectories $\tau^{ji} = \{(s_t^{ij}, u_t^{ij}) : t = 1 : T\}, j = 1, \cdots, N$ from

$$q_i(s_t^{ij}, u_t^{ij}) = p_{i1}(s_t^{ij}) \prod_{t=1}^{T} q_i(u_t|s_t)p(s_{t+1}|s_t, u_t),$$

induced by each of the guiding distributions $q_i, i = 1, \cdots, M$ with $p_{i1}(s_t^{ij}), i = 1, \cdots, M$ being the initial state distribution.

**Step 3:** Train the NN representing the policy mean $\mu^\pi(s; w)$ with a modified objective (8) which under Gaussian assumption becomes:

$$\arg\min_{w} \frac{1}{2N} \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[ \nu_t(\mu^\pi(s_t^{ij})) - \mu^\pi(s_t^{ij}) - \mu^\pi(s_t^{ij}; w) + 2\lambda_{\mu}^\pi(\mu^\pi(s_t^{ij}; w)) \right].$$

The policy covariance $\Sigma^\pi$ can be computed directly as $\Sigma^\pi = \left[ \frac{1}{N} \sum_{t=1}^{T} \sum_{i=1}^{M} (\Sigma^\pi)^{-1} \right]^{-1}$.

**Step 4:** Update the Lagrange multipliers

$$\lambda^\mu_{it} \leftarrow \lambda^\mu_{it} + \alpha\nu_t \frac{1}{N} \sum_{j=1}^{N} [\mu^\pi(s_t^{ij}; w) - \mu^\pi(s_t^{ij})],$$

for $t = 1, \cdots, T \quad i = 1, \cdots, M$.

### A. Adaptation of GPS Framework

Application of GPS framework to the CS control problem stated in Section II required further adaptations. For the selected CS application, achieving the desired profile required multiple passes. Hence, intermediate goals were defined for computing iLQR based local optimal controllers which will depend on part/defect geometry. Figure 2b shows examples of intermediate goal profiles $d^i_j$ for the test part considered in our work (see Section IV-A) which were determined so that in each pass least amount of build up happens on the flat portion and maximum deposition occurs in the notch. The objective for the $i$-th iLQR was to achieve the desired goal profile $d^i_j$ starting from the previous goal $d^{i-1}$. Furthermore, the objective function in (4) was modified for each $i$-th iLQR with this goal profile, and the terminal cost term $c_x$ was absorbed in the total cost leading to $\sum_{t=1}^{T}|c(s_t, u_t)| + |d_t - d^i_j|^2$. In order to generate training data for NN, initial starting surface for each iLQR was perturbed which not only involved randomized to account for measurement noise in the experiments [20], but also included the propagation effect of previous iLQR controller in achieving their goal surface profiles. Generation of rich training sets which covers different possibilities of intermediate states is necessary in sequential decision making tasks: as if NN control actions resulted in surface buildups never encountered in the training, that will further drive the NN to make errors resulting in cascading failure. We will refer to NN controller trained using GPS as GPS based controller (GPS).

### IV. RESULTS

#### A. Experimental Setup

The control setup for GPSC demonstration comprised of (see Fig. 3): CS machine/robot, laser displacement sensor (LDS), and standard PC to host the GPSC algorithm. The CS machine comprised of gas heater, powder feeder, and nozzle mounted on a 5-axis robot. A turn table was used to rotate the part to be coated, and thus requiring only a linear motion of nozzle for deposition on the complete surface of an axis symmetric part. The LDS had a 640 pixel resolution with a sampling rate of 1kHz. LABVIEW was used for implementing the control logic which sequentially activated the LDS to provide surface profile measurements, followed by the GPSC code for nozzle speed computation based on the measurements, and then transmitted the nozzle speed commands to the robot controller. For testing the GPSC framework, we used an aluminium cylindrical test coupon with a symmetrical notch representing a defect to be repaired as shown in Fig. 3b.

#### B. Model Calibration

We used an optimization framework for CS model parameter calibration. The calibration data was generated by making several consecutive passes on the test coupon, where for each pass the speed was held constant taking values $v^i_t \in \{1, 2, 3, 4, 5\}$mm/sec. The surface profiles after each pass were recorded using the LDS, which we denote by $d_t, i = 0, \cdots, N_c$. Let $d_t(p; d_{i-1})$ be predicted surface using the CS model (3) starting with $d_{i-1}$, and using parameters $p$ and constant nozzle speed $\{v_t^i\}_{t=0}^T$ over the pass. The optimization problem can be defined as (see Fig. 4a)

$$\min_{p} \sum_{i=1}^{N_c} ||d_{i}(p; d_{i-1}) - d_{i}||^2.$$  \hfill (10)

We additionally introduced linear constraints $p_{\text{min}} \leq p \leq p_{\text{max}}$ to bound the parameter values in a desired range. The optimization (10) was solved using fmincon nonlinear optimization routine in MATLAB using a subset of training data, and the remaining set was used for validation. Figure 4b-c shows the calibration results on training and the validation dataset, respectively. The black curves represents experimental data while dotted red curves show calibrated model predictions.

#### C. GPSC Deployment

We used the GPSC framework described in Section III-A, where only the nozzle speed $v^i_t$ was the controlled variable, the nozzle height $h_t$ and orientation $\alpha_t$ were kept fixed. We used $N_d = 100$ for the discretized representation of the surface (see Section IV-A). Consequently, the input layer of the NN consisted of 101 neurons (with one additional input for the nozzle position), and output layer had one neuron corresponding to the nozzle speed. We found that 1 hidden layer with 10 neurons was sufficient for our application. In order to ensure that the output speed remains bounded in range $[0 \, 5]$, we selected a sigmoid activation function for
the output layer. The closed loop control was implemented at every pass, i.e. after measurement from LDS was available the controller computed nozzle speed for the entire next pass by propagating the calibrated CS model. This setting of feedforward/feedback control is expected to commonly arise in AM applications, where meaningful sensing data may be only available after one or many layers have been deposited. For repetitive spraying, a cyclic robot motion plan was created, which consisted of the two sets of waypoints (see Fig. 3 b): $P_{10}, P_{20}, P_{30}, P_{40}$ are the teach waypoints and remain fixed, while the control waypoints $W_1, \ldots, W_N$ depended on the nozzle speed control commands generated by GPSC.

D. Findings

We compare performance of GPSC with two baseline open loop control strategies: constant speed and varying speed. In both these cases the nozzle speed is not changed based on the in-situ LDS measurements. For the constant speed case the speed of nozzle is set to be a constant value of 1 mm/sec. For varying speed case the speed changes with nozzle location to prescribed values as shown in Fig. 5c. This speed profile was hand crafted by expert so that nozzle moves fastest in flat portion of coupon resulting in minimal deposition and moves slower in the notch area which requires more deposition. The nozzle speed profiles generated for the different passes during GPSC is shown in the Fig. 5b. Note that GPSC speed profiles are initially similar to the varying speed profile, but adapt as more layers were build to account for various sources of uncertainties/disturbances including robots precision in following the commanded speed, wobbling of the cylinder during rotation, and the variations in powder characteristics. Due to lack of this adaptation in the open loop varying speed case, the final deposit was significantly irregular (see Figure 5a). The irregular deposition occurs around start and end of the notch, as the robot controller does not exactly follow the speed profile around those transition points. The robot slows down and increases speed sooner than prescribed, leading to over and under deposition around the start and end of the notch, respectively. Overall, GPSC saves 16% scrap material over constant speed, and around 35% over varying speed profile. Furthermore, GPSC terminated in 33 min compared to 38 min for varying speed and 32 min for constant speed cases: thereby resulting in time savings as well.

V. CONCLUSIONS

In this paper we applied guided policy search based RL approach for a high dimensional nonlinear optimal control problem arising in cold spray manufacturing process. The approach was experimentally validated and showed promising performance. In future, we plan to explore hierarchical
RL framework [21] for automatically learning hierarchical structure in additive manufacturing control problems which has a potential to further improve generalization and transfer to different part geometries.

ACKNOWLEDGMENT

Funding provided by RTRC is greatly appreciated. Authors will like to thank Matthew Siopis for supporting the experimental demonstrations.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[2] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in International conference on machine learning, 2016, pp. 1928–1937.
[3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play,” Science, vol. 362, no. 6419, pp. 1140–1144, 2018.
[4] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 1334–1373, 2016.
[5] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” arXiv preprint arXiv:1509.02971, 2015.
[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv preprint arXiv:1707.06347, 2017.
[7] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, “Benchmarking deep reinforcement learning for continuous control,” CoRR, vol. abs/1606.06778, 2016. [Online]. Available: http://arxiv.org/abs/1606.06778
[8] S. Levine and V. Koltun, “Guided policy search,” in Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, 2013, pp. 1–9.
[9] J. Peters and S. Schaal, “Reinforcement learning of motor skills with policy gradients,” Neural networks, vol. 21, no. 4, pp. 682–697, 2008.
[10] J. Kober, A. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” The International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.
[11] S. Yin, P. Cavaliere, B. Aldwell, R. Jenkins, H. Liao, W. Li, and R. Lupoi, “Cold spray additive manufacturing and repair: Fundamentals and applications,” Additive Manufacturing, vol. 21, pp. 628–650, 2018.
[12] G. Tapia and A. Elwany, “A review on process monitoring and control in metal-based additive manufacturing,” Journal of Manufacturing Science and Engineering, vol. 136, no. 6, pp. 060801, 2014.
[13] S. K. Everton, M. Hirsch, P. Stravroulakis, R. K. Leach, and A. T. Clare, “Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing,” Materials & Design, vol. 95, pp. 431–445, 2016.
[14] X. Qi, G. Chen, Y. Li, X. Cheng, and C. Li, “Applying neural-network-based machine learning to additive manufacturing: Current applications, challenges, and future perspectives,” Engineering, 2019.
[15] A. Chakraborty, S. Shishkin, and M. J. Birnkraut, “Optimal control of build height utilizing optical profilometry in cold spray deposits,” in Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2017, vol. 10168. International Society for Optics and Photonics, 2017, p. 101683H.
[16] A. Surana, “Reinforcement learning: An industrial perspective,” in Handbook of Reinforcement Learning and Control. Springer, 2021, pp. 647–672.
[17] X. Wang, F. Feng, M. A. Klecka, M. D. Mordasky, J. K. Garofano, T. El-Wardany, A. Nardi, and V. K. Champagne, “Characterization and modeling of the bonding process in cold spray additive manufacturing,” Additive Manufacturing, vol. 8, pp. 149–162, 2015.
[18] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems,” in ICINCO (1), 2004, pp. 222–229.
[19] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, “Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search,” CoRR, vol. abs/1509.06791, 2015.
[20] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 23–30.
[21] O. Nachum, S. S. Gu, H. Lee, and S. Levine, “Data-efficient hierarchical reinforcement learning,” in Advances in Neural Information Processing Systems, 2018, pp. 3303–3313.