2023 9th International Conference on Control, Decision and Information Technologies
CoDIT 2023 | Rome, Italy / July 03-06, 2023 Technically co-sponsored by IEEE & IFAC

Output Feedback Reinforcement Learning for Temperature Control in a Fused Deposition Modelling Additive Manufacturing system

Eleni Zavrakli*[1,2,3] , Andrew Parnell [1,2,3] and Subhrakanti Dey[4]

2023 9th International Conference on Control, Decision and Information Technologies (CoDIT) | 979-8-3503-1140-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/CODIT58514.2023.10284168

***Abstract*— The development of effective closed-loop control algorithms is one of the main challenges in Additive Manufacturing (AM). Many parameters of AM processes need continuous monitoring and regulation, with temperature being one of the most important. We investigate the design of an output-feedback controller of the temperature process within the extruder of a Fused Deposition Modelling (FDM) AM system. Based on a state space approach, and using input-output measurements, we first design a model-based linear quadratic tracking controller, followed by an equivalent model-free, data-driven version. We demonstrate these approaches using a simulator of the temperature evolution in the extruder of the AM system, based on a model validated and identified in recent literature. Our findings show that a comparable performance to the model-based case is possible using only measured data, generated through probing control explorations during the simulations.**

I. INTRODUCTION

The need for designing closed-loop controllers for Additive Manufacturing (AM) is becoming increasingly evident with the wider adoption of AM in many aspects of science, technology and daily life [1]. Classic control theory [2], [3] offers a variety of reliable methods to optimise the system's behaviour under different criteria, mainly when a model of the system's dynamics is available. In the sub-field of Dynamic Programming [4], [5], a family of controllers is framed as decision-making problems. Reinforcement Learning (RL) [6] is the equivalent area of machine learning that deals with decision-making problems, often with insufficient model knowledge. These approaches are initially grounded in information obtained from a model of the system but are adapted to become model-free with the use of process data to learn and approximate the optimal policies [7]. Due to its ability to learn efficient controllers without the need of a process model, RL is a promising candidate for the design of closed-loop control for AM, considering that many AM processes are too complicated to derive exact models.

In this work we design tracking controllers for the temperature within the extruder head of a Big Area Additive Manufacturing (BAAM) system [8]. The manufacturing method used by the system is Fused Deposition Modelling (FDM), also known as Material Extrusion (MEX). We base our control approach on the output feedback reinforcement learning algorithm introduced in [9]. Similar methods have been proposed in works such as [10], [11]. We establish the controller design using the system model and later explore the case where no model is available, instead the control design needs to be determined through data. We assume access to a record of past input and output measurements but not the internal state of the system. This record is generated during simulations by using persistently exciting inputs.

In Section II we introduce the problem as the Linear Quadratic Tracking (LQT) problem for a state space system. In Section III we study the solution to the LQT problem using Reinforcement Learning. Section IV focuses on solving the LQT problem using past input and output data and in Section V this problem is approached in a model-free data-driven fashion, assuming no knowledge of a system model. Finally, in Section VI we present the setup and results of our simulation experiments and compare the performance of the model-based and data-driven controllers.

II. LINEAR QUADRATIC TRACKING CONTROL FOR AN MATERIAL EXTRUSION STATE SPACE MODEL

Uncontrolled temperature variation is one of the main causes of defects in AM. It affects many different aspects of the process such as the melting of the material, the binding between layers, the solidification of the printed object and how well the object is attached to the build plate during the printing. We specifically focus on the temperatures within the extruder of FDM systems. We adopt a discrete-time linear state space model introduced in [8] to design a model-based LQT controller. We also use the model to simulate data to train the data-driven controller. This model was validated through system identification methods using input and response data from a BAAM system in [8]. The extruder is made up of five main parts: the hopper, the screw, the barrel, the hose and the nozzle. There are six heaters in total providing thermal energy to the system and an AC motor which rotates the screw. Four of the heaters are located in the barrel, one in the hose and one in the nozzle. Each heater can also be considered a thermal cell, whose temperature is influenced by the neighbouring cells and the motor. The system can be expressed as a linear model in state space form

$$x(t+1) = Ax(t) + Bu(t),\ t \geq t_0 \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the system state vector at discrete time point t and $u(t) \in \mathbb{R}^m$ is the system input at t. $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the state and input matrices respectively. In the

*Corresponding author: eleni.zavrakli@mu.ie
[1]Department of Mathematics and Statistics, Maynooth University, Co. Kildare, Ireland
[2]I-Form Advanced Manufacturing Research Centre, Ireland
[3]Hamilton Institute, Maynooth University, Co. Kildare, Irealand
[4]Division of Signals and Systems, Department of Electrical Engineering, Uppsala University, Sweden

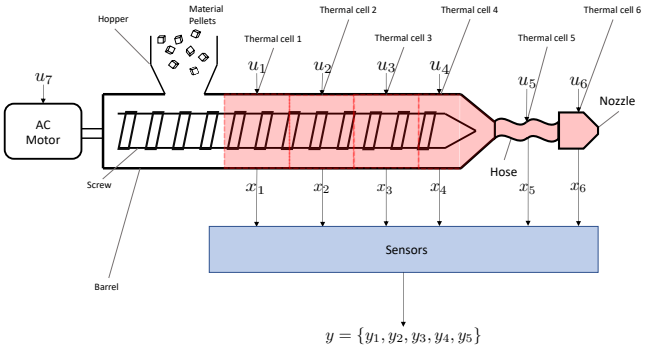

Fig. 1. Heating zones in the BAAM extruder. Four of the zones are located in the barrel, heating the material as it gets pushed by the screw towards the hose. The heating continues in the hose and finally in the nozzle, through which the melted material gets distributed.

case of our system, $x(t) \in \mathbb{R}^6$ is the set of the temperatures in each thermal cell and $u(t) \in \mathbb{R}^7$ is the input provided by the heaters and the motor, for each time point t.

Additionally, we assume that $x(t)$ is not directly available. Instead, a set of measurements is obtained through

$$y(t) = Cx(t),\ t \geq t_0 \tag{2}$$

where $y(t) \in \mathbb{R}^p$ is the system output at time t and $C \in \mathbb{R}^{p \times n}$ is the measurement matrix. We assume that (A,B) is controllable and (A,C) is observable. Figure 1 provides a visual representation of the heating system in the extruder of the BAAM system.

The optimal tracking problem is defined as the search for a control function u^* such that the system output accurately tracks the reference signal r generated by

$$r(t+1) = Fr(t) \tag{3}$$

where F is the reference system matrix. The optimisation objective can be mathematically expressed as the minimisation of the performance index

$$\begin{aligned} V(x,r,u) &= \sum_{t=k}^{\infty} \gamma^{t-k} \big\{ [y(t) - r(t)]^T Q [y(t) - r(t)] \\ &\quad + u^T(t) R u(t) \big\} \\ &= \sum_{t=k}^{\infty} \gamma^{t-k} \big\{ [Cx(t) - r(t)]^T Q [Cx(t) - r(t)] \\ &\quad + u^T(t) R u(t) \big\} \end{aligned} \tag{4}$$

where $Q \in \mathbb{R}^{p \times p}$ is the tracking error weighting matrix and $R \in \mathbb{R}^{m \times m}$ is the input weighting matrix. $0 < \gamma \leq 1$ is a discount factor, whose role is to weigh short-term costs more heavily than costs in the distant future. In the RL literature, V is also referred to as the value function.

III. Model-based Reinforcement Learning Solution to the Optimal Tracking Problem

We create the augmented state by attaching the reference to the system state,

$$X(t) = \begin{bmatrix} x(t) \\ r(t) \end{bmatrix} \tag{5}$$

and construct the augmented system state equation

$$\begin{aligned} X(t+1) &= \begin{bmatrix} x(t+1) \\ r(t+1) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & F \end{bmatrix} \begin{bmatrix} x(t) \\ r(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) \\ &:= TX(t) + B_1 u(t). \end{aligned} \tag{6}$$

where T and B_1 are defined to be the augmented system state and input matrices respectively.

The value function 4 can be written in terms of the augmented state as

$$V(X,u) = \sum_{t=k}^{\infty} \gamma^{t-k} \{ X^T(t) Q_1 X(t) + u^T(t) R u(t) \} \tag{7}$$

where

$$Q_1 = \begin{bmatrix} C^T Q C & -C^T Q \\ -QC & Q \end{bmatrix}. \tag{8}$$

The solution to the optimal tracking problem is a policy of the form

$$u^*(t) = -KX(t). \tag{9}$$

Lemma 1: [7] For the optimal tracking problem with value function of the form 4 and reference of the form 3 then for any stabilising policy of the form 9, the value function can be written in quadratic form as

$$V(x,r,u) = V(X) = \frac{1}{2} X^T(t) P X(t) \tag{10}$$

for some matrix $P = P^T > 0$.
It was shown in [7] that the optimal control policy K in 9 can be obtained through

$$K = (R + \gamma B_1^T P B_1)^{-1} \gamma B_1^T P T \tag{11}$$

where P is the solution to the augmented Algebraic Riccati Equation (ARE)

$$Q_1 - P + \gamma T^T P T - \gamma^2 T^T P B_1 (R + \gamma B_1^T P B_1)^{-1} B_1^T P T = 0. \tag{12}$$

Solving an ARE directly can be a computationally complex task. Instead, consider the quadratic form of the value function in terms of the augmented state $X(t)$ 10 and the corresponding form of the performance index 7. We can obtain the LQT Bellman equation

$$\begin{aligned} X^T(t) P X(t) &= X^T(t) Q_1 X(t) + u^T(t) R u(t) \\ &\quad + \gamma X^T(t+1) P X(t+1). \end{aligned} \tag{13}$$

Assuming a specific stabilizing policy K in 13, yields the Lyapunov equation

$$P = Q_1 + K^T R K + \gamma (T + B_1 K)^T P (T + B_1 K). \tag{14}$$

Iteratively alternating between solving the Lyapunov equation and determining the control policy through 9 results in the optimal solution to the LQT problem.

However, not having access to the system state x presents an important limitation. When the model is available and the system is fully observable, this limitation can be overcome with the design of a state observer [12]. When designing a model-free controller a different approach needs to be taken.

IV. Output Feedback using Input-Output data

Following the approach introduced in [9], we assume access to past input and output sequences and past reference signals for a time horizon N and can write the state as follows:

$$\begin{aligned} x(t) &= A^N x(t-N) \\ &\quad + \begin{bmatrix} B & AB & A^2B & \dots & A^{N-1}B \end{bmatrix} \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-N) \end{bmatrix} \\ &:= A^N x(t-N) + U_N \bar{u}(t-1,t-N) \end{aligned} \tag{15}$$

Then the system output can be written as

$$y(t) = Cx(t) = CA^N x(t-N) + CU_N \bar{u}(t-1,t-N) \tag{16}$$

Using the above, the sequence of outputs for the time horizon $[t-N,t-1]$ can be expressed as

$$\begin{aligned} \bar{y}(t-1,t-N) &= \begin{bmatrix} y(t-1) \\ y(t-2) \\ y(t-3) \\ \vdots \\ y(t-N) \end{bmatrix} = \begin{bmatrix} CA^{N-1} \\ CA^{N-2} \\ \vdots \\ CA \\ C \end{bmatrix} x(t-N) \\ &\quad + \begin{bmatrix} 0 & CB & CAB & \dots & CA^{N-2}B \\ 0 & 0 & CB & \dots & CA^{N-3}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & CB \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-N) \end{bmatrix} \\ &:= W_N x(t-N) + D_N \bar{u}(t-1,t-N) \end{aligned} \tag{17}$$

Since the pair (A,C) is observable, the matrix W_N has full rank for any choice of $N \geq K$ where K is the observability index. Using the generalised inverse $W_N^+ = (W_N^T W_N)^{-1} W_N^T$, the augmented state 5 can be expressed as

$$\begin{aligned} X(t) = & \begin{bmatrix} A^N & 0 \\ 0 & F^N \end{bmatrix} \begin{bmatrix} W_N^+ (\bar{y}(t-1,t-N) - D_N \bar{u}(t-1,t-N)) \\ r(t-N) \end{bmatrix} \\ & + \begin{bmatrix} U_N \\ 0 \end{bmatrix} \bar{u}(t-1,t-N) \\ = & \begin{bmatrix} U_N - A^N W_N^+ D_N & A^N W_N^+ & 0 \\ 0 & 0 & F^N \end{bmatrix} \begin{bmatrix} \bar{u}(t-1,t-N) \\ \bar{y}(t-1,t-N) \\ r(t-N) \end{bmatrix} \end{aligned}$$

or in a more compact form:

$$X(t) = \begin{bmatrix} x(t) \\ r(t) \end{bmatrix} = M \begin{bmatrix} \bar{u}(t-1,t-N) \\ \bar{y}(t-1,t-N) \\ r(t-N) \end{bmatrix} \tag{18}$$

where $M = \begin{bmatrix} U_N - A^N W_N^+ D_N & A^N W_N^+ & 0 \\ 0 & 0 & F^N \end{bmatrix}$.

With the use of the augmented system dynamics 6, the Bellman equation 13 can be written as

$$V(Z) = \frac{1}{2} Z^T(t) H Z(t) \tag{19}$$

with $Z(t) = \begin{bmatrix} X(t) \\ u(t) \end{bmatrix}$ and $H = \begin{bmatrix} Q_1 + \gamma T^T P T & \gamma T^T P B_1 \\ \gamma B_1^T P T & R + \gamma B_1^T P B_1 \end{bmatrix}$. Using 18 in 19 we obtain

$$\begin{aligned} V(Z) &= \frac{1}{2} \begin{bmatrix} X(t) \\ u(t) \end{bmatrix}^T H \begin{bmatrix} X(t) \\ u(t) \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} M \begin{bmatrix} \bar{u}(t-1,t-N) \\ \bar{y}(t-1,t-N) \\ r(t-N) \end{bmatrix} \\ u(t) \end{bmatrix}^T H \begin{bmatrix} M \begin{bmatrix} \bar{u}(t-1,t-N) \\ \bar{y}(t-1,t-N) \\ r(t-N) \end{bmatrix} \\ u(t) \end{bmatrix} \end{aligned}$$

Defining $\bar{Z}(t) = \begin{bmatrix} \bar{u}(t-1,t-N) \\ \bar{y}(t-1,t-N) \\ r(t-N) \\ u(t) \end{bmatrix}$ and $\bar{H} = \begin{bmatrix} M & 0 \\ 0 & I_{m\times m} \end{bmatrix}^T H \begin{bmatrix} M & 0 \\ 0 & I_{m\times m} \end{bmatrix}$, we can rewrite the value function in the form

$$\begin{aligned} V(\bar{Z}) := \frac{1}{2} \bar{Z}^T(t) \bar{H} \bar{Z}(t) = \frac{1}{2} & \begin{bmatrix} \bar{u}(t-1,t-N+1) \\ \bar{y}(t,t-N+1) \\ r(t-N+1) \\ u(t) \end{bmatrix}^T \\ & \begin{bmatrix} H_{\bar{u}\bar{u}} & H_{\bar{u}\bar{y}} & H_{\bar{u}r} & H_{\bar{u}u} \\ H_{\bar{y}\bar{u}} & H_{\bar{y}\bar{y}} & H_{\bar{y}r} & H_{\bar{y}u} \\ H_{r\bar{u}} & H_{r\bar{y}} & H_{rr} & H_{ru} \\ H_{u\bar{u}} & H_{u\bar{y}} & H_{ur} & H_{uu} \end{bmatrix} \begin{bmatrix} \bar{u}(t-1,t-N+1) \\ \bar{y}(t,t-N+1) \\ r(t-N+1) \\ u(t) \end{bmatrix} \end{aligned} \tag{20}$$

which gives rise to the Bellman equation

$$\begin{aligned} \bar{Z}(t)^T \bar{H} \bar{Z}(t) &= (y(t) - r(t))^T Q (y(t) - r(t)) + u^T(t) R u(t) \\ &\quad + \gamma \bar{Z}(t+1)^T \bar{H} \bar{Z}(t+1). \end{aligned} \tag{21}$$

Applying the optimality condition $\frac{\partial V}{\partial u} = 0$ and solving for $u(t)$ yields

$$\begin{aligned} u(t) &= -H_{uu}^{-1} (H_{u\bar{u}} \bar{u}(t-1,t-N) \\ &\quad + H_{u\bar{y}} \bar{y}(t-N,t-N) + H_{ur} r(t-N)). \end{aligned} \tag{22}$$

If the system model is available, 22 can be used to directly determine the optimal LQT controller using input-output information. The kernel matrix $\bar{H}$ can be obtained using its definition and 19, where P is obtained by solving 14for the standard LQT problem.

V. Model-free Output Feedback

When a reliable model of the system is not available, namely matrices A, B and C are unknown, the kernel matrix $\bar{H}$ needs to be estimated using measured data. This can be achieved using the Value Iteration (VI) algorithm. It involves solving the Bellman equation 21 for kernel matrix $\bar{H}$. To that end, $\bar{H}$ needs to be isolated from the quadratic form. This can be achieved by firstly vectorising the Bellman equation 21

$$\begin{aligned} vec\left(\bar{Z}^T(t) \bar{H} \bar{Z}(t)\right) &= (y(t) - r(t))^T Q (y(t) - r(t)) + u^T(t) R u(t) \\ &\quad + \gamma vec\left(\bar{Z}^T(t+1) \bar{H} \bar{Z}(t+1)\right) \end{aligned}$$

and then applying the "vector trick" associated with the Kronecker product

$$\left(\bar{Z}^T(t)\otimes\bar{Z}^T(t)\right)vec(\bar{H}) = (y(t)-r(t))^T Q(y(t)-r(t)) + u^T(t)Ru(t)+\gamma\left(\bar{Z}^T(t+1)\otimes\bar{Z}^T(t+1)\right)vec(\bar{H}). \quad (23)$$

For the implementation of the VI algorithm, we design an initial kernel matrix $\bar{H}^0$ that obtains an initial admissible policy $u^0(t)$, using measured data [13]. We then iterate between the following two steps:

1) **Policy Evaluation**

$$\left(\bar{Z}^T(t)\otimes\bar{Z}^T(t)\right)vec(H^{\bar{i}+1}) = (y(t)-r(t))^T Q(y(t)-r(t)) + (u^i)^T(t)Ru^i(t)+\gamma\left(\bar{Z}^T(t+1)\otimes\bar{Z}^T(t+1)\right)vec(\bar{H}^i)$$

2) **Policy Improvement**

$$u^{i+1}(t) = -(H_{uu}^{i+1})^{-1}(H_{u\bar{u}}\bar{u}^{i+1}(t-1,t-N) + H_{u\bar{y}}^{i+1}\bar{y}(t-1,t-N)+H_{ur}^{i+1}r(t-N))$$

The Policy evaluation step can be solved through the Least Squares (LS) algorithm using measured data $\bar{Z}(t),\bar{Z}(t+1)$ and calculating the cost term $(y(t)-r(t))^T Q(y(t)-r(t))+(u^i)^T(t)Ru^i(t)$ for each data point using the current control function estimate u^i. Matrix $\bar{H}$ is an $((N+1)m+(N+1)p)\times((N+1)m+(N+1)p)$ symmetric matrix which means that its determination is a problem with $((N+1)m+(N+1)p)\times((N+1)m+(N+1)p)/2$ degrees of freedom. This is also the minimum number of data points needed for the VI algorithm, but in practice many more data points are usually needed, especially when dealing with more complicated systems with larger state and action spaces. When the data are highly correlated, which is often the case with sequential data, the inversion step in the LS algorithm becomes challenging. This can be rectified with a regularized LS approach, with an appropriate regularisation parameter μ.

VI. Simulation Setup and Results

The model we use for our simulations and the design of the data-driven controller is the state space model obtained in [8] using system identification methods. We use the state and input matrices A and B as determined in [8]

$$A = \begin{bmatrix} 0.992 & 0.0018 & 0 & 0 & 0 & 0 \\ 0.0023 & 0.9919 & 0.0043 & 0 & 0 & 0 \\ 0 & -0.0042 & 1.0009 & 0.0024 & 0 & 0 \\ 0 & 0 & 0.0013 & 0.9979 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9972 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9953 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0033 & 0 & 0 & 0 & 0 & 0 & -0.2175 \\ 0 & 1.0460 & 0 & 0 & 0 & 0 & -0.0788 \\ 0 & 0 & 1.0326 & 0 & 0 & 0 & -0.0020 \\ 0 & 0 & 0 & 0.4798 & 0 & 0 & -0.0669 \\ 0 & 0 & 0 & 0 & 0.8882 & 0 & 0.1273 \\ 0 & 0 & 0 & 0 & 0 & 1.1699 & -0.1792 \end{bmatrix}$$

and we design an output matrix

$$C = \begin{bmatrix} 0.992 & 0.00018 & 0 & 0 & -0.0001 & 0 \\ 0.0023 & 1.3 & 0.0043 & 0 & 0 & 0 \\ 0 & -0.0042 & 1.0109 & 0.0024 & 0 & 0.201 \\ 0 & 0 & 0.0013 & 0.989 & 0.00031 & 0.64 \\ 0 & 0 & 0 & 0 & 0.923 & 0.3 \end{bmatrix}$$

where we assume we obtain 5 measurements out of the 6 states. As needed, we verified that (A,B) is controllable and (A,C) is observable.

We chose the goal of our optimisation to be bringing and maintaining all system states at some predefined value. We chose that value to be 180 °C, which is within the melting temperature range for the material used in the AM system, meaning that the reference trajectory is defined as $r(t) = \{180,\ldots,180\}\in\mathbb{R}^5$ for all time points t and the reference generator matrix is $F = I_6$. For the weighting matrices we chose identity matrices of appropriate dimensions $Q = I_6$ and $R = I_7$. The initial state is chosen arbitrarily to be $x(t_0) = \{50,\ldots,50\}\in\mathbb{R}^6$ and the discount factor $\gamma = 0.99$. The time horizon for the input and output sequences is chosen to be equal to the observability index of the system which is $N = 6$.

Figure 2 shows the trajectories of the measured temperatures within 100 time steps, after applying the model-based controller designed using input-output data 22. The output converges to the vector $y^* = (179.98, 180, 180, 179.99, 179.99)$ rounded up in two decimal places, and arrives within 0.1 degrees or 0.06% of the reference in 17 steps. This performance is expected and justified, given that a model that perfectly describes the system dynamics is available and can be utilised for the controller design.

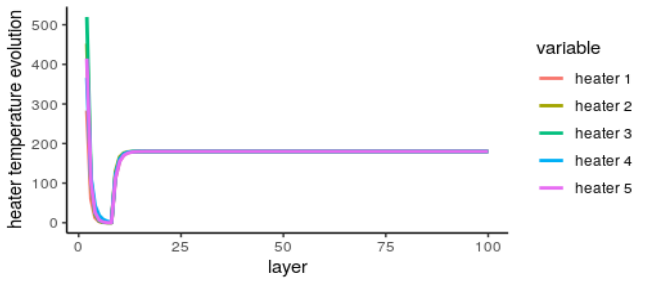

Fig. 2. Optimal trajectories. When the system dynamics are known, the controller successfully brings and maintains the temperatures within a very small margin of the optimal of 180 degrees after 11 time steps.

For the data-driven controller, we chose the same values for the shared parameters as in the previous case so we can make reasonable comparisons. We generate data by applying persistently exciting inputs to the model introduced above. To design such inputs we include a noise term formulated as the sum of sinusoidal functions of varying frequencies and amplitudes. A large amount of data needs to be generated to efficiently train the algorithm. Specifically we produce 13,000 data points. The data produced are appropriately normalized for numerical stability. The regularisation parameter for the LS algorithm is chosen to be $\mu = 0.01$ and we iterate for 1000 iterations at a time, or until two consecutive estimates of $\bar{H}$ are within 0.001 of each other. We obtain a controller that makes the system outputs converge to the vector $y^* = (189.99, 186.94, 185.3, 183.78, 187.12)$. The trajectories converge to final values that are within 6% of the optimal value of 180. The values reach and remain within that error window in 11 steps. Figure 3 shows the

trajectories of the outputs when applying the model-free controller obtained through the VI algorithm.

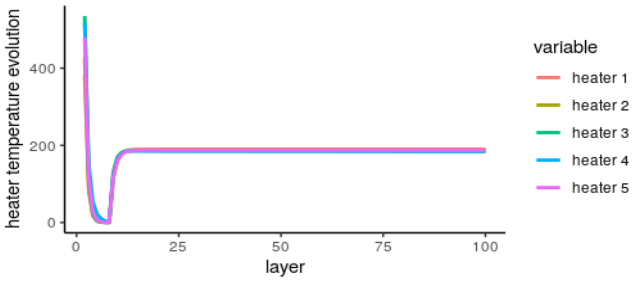

Fig. 3. Trajectories obtained when the data-driven controller is applied to the system.

VII. Conclusion

We have studied the problem of designing controllers for the temperature in a BAAM system using input-output data. Using Reinforcement Learning, an effective Linear Quadratic Tracker can be designed from the model of the system by augmenting the feedback term to use a record of past inputs, outputs and references. We then focused on solving the same problem in a data-driven way, based on input and output measurements only, with no knowledge of a model. We found that we can achieve a performance that is close to the optimal through using persistently exciting inputs for data generation, normalising the data and using a regularisation parameter.

Future research efforts will focus on obtaining a performance closer to the model-based results by optimising the choice of parameters in the simulation setup and specifically the weighting matrices Q and R. While there seems to be few systematic approaches to optimise the choice of these matrices, some learning algorithms such as Bayesian Optimisation or Evolutionary Algorithms may be used. Another result that needs to be improved is the overshoot in the final trajectories, even in the case of the model-based controller. This issue can be addressed with the use of constrains in the performance index, which would result in a non-quadratic form. In this case, Deep Neural Networks can be used to approximate the value function and generate new optimal policies, also known as the actor-critic framework.

References

[1] F. J. Mercado Rivera and A. J. Rojas Arciniegas, "Additive manufacturing methods: techniques, materials, and closed-loop control applications," *The International Journal of Advanced Manufacturing Technology*, vol. 109, pp. 17–31, 2020.

[2] B. D. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[3] K. Ogata *et al.*, *Modern control engineering*. Prentice hall Upper Saddle River, NJ, 2010, vol. 5.

[4] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific, 1996.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, 2014.

[8] D. Gootjes, "Applying feedback control to improve 3d printing quality," Master's thesis, Delf University of Technology, 2017.

[9] C. Chen, W. Sun, G. Zhao, and Y. Peng, "Reinforcement q-learning incorporated with internal model method for output feedback tracking control of unknown linear systems," *IEEE Access*, vol. 8, pp. 134 456–134 467, 2020.

[10] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE transactions on cybernetics*, vol. 45, no. 12, pp. 2770–2779, 2015.

[11] S. A. A. Rizvi and Z. Lin, "Output feedback optimal tracking control using reinforcement q-learning," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 3423–3428.

[12] D. Luenberger, "An introduction to observers," *IEEE Transactions on automatic control*, vol. 16, no. 6, pp. 596–602, 1971.

[13] V. G. Lopez, M. Alsalti, and M. A. Müller, "Efficient off-policy q-learning for data-based discrete-time lqr problems," *IEEE Transactions on Automatic Control*, 2023.