SAND2022-1704C

Constrained Run-to-Run Control for Precision Serial Sectioning

Damian Gallegos-Patterson[†‡], Kendric Ortiz[†‡], Jonathan Madison[‡], Andrew Polonsky [‡] Claus Danielson[†]

***Abstract*— This paper presents a run-to-run (R2R) controller for mechanical serial sectioning (MSS). MSS is a destructive material analysis process which repeatedly removing a thin layer of material and imaging the exposed surface. The images are then used to construct a 3-dimensional image of a material sample. Currently, an experience human operator selects the parameters of the MSS to achieve the desired thickness. The proposed R2R controller will automate this process while improve the precision of the material removal. The proposed R2R controller solves an optimization problem designed to minimize the variance of the material removal subject achieving the expected target removal. This optimization problem was embedded in an R2R framework to provide iterative feedback for disturbance rejection and convergence to the desired removal rate. Since an analytic model of the MSS system is unavailable, we adopted a data-driven approach to synthesize our R2R controller from historical data. The proposed R2R controller is demonstrated through simulations. Future work will empirically demonstrate the proposed R2R through experiments with a real MSS system.**

I. Introduction

Serial sectioning is a destructive analysis process used to gain insight on the micro-structure of materials. Serial section is typically used when non-destructive analysis is insufficient, due either to material composition or a higher resolution of data for a specific features is required. Serial sectioning repeatedly removes thin layers of material and captures a detailed image of the exposed surface. At the end of a serial sectioning experiment, this sequence of images are combined to create a 3D reconstruction of the sample. The 3D reconstruction can be used to gain knowledge of the size, shape, and location of features of interest within the overall sample. This process is used for failure analysis, feature identification, and material composition experiments [1]. A common use is identification of flaws in 3D printed metal samples to locate the size and location of areas of the sample which leftover powered metal may still exist or where the print may have left a large void. There are three different types of serial sectioning tools, micro-mill serial sectioning, FIB laser serial sectioning, and mechanical serial sectioning (MSS). This paper will be focusing on a MSS process.

This material is based upon work supported by the National Science Foundation under NSF Grant Number CNS-2105631. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. [†]University of New Mexico. [‡]Sandia National Laboratories. Corresponding Author: Damian Gallegos-Patterson dagallegospatterson@unm.edu

MSS uses a three phase repetitive process of grinding, polishing, and optical imagining, to collect data from a material sample. Each cycle of polishing and optical imaging is called a *slice* which produces a montage of images which can be stitched together to produce a larger cross-sectional image of the entire surface or a specific region of interest. The human operator inputs the desired number of slices which is called a *run*. A run consists of the human operator selecting a *recipe* a sequence of grinding pads and polishing pads, as well as the polish time, speed, solution, and solution dispensing time. The parameters of this recipe are the inputs to the MSS system. This paper will develop an autonomous controller for iteratively selecting the appropriate recipe to achieve a target material removal amount. The current method for achieving the target removal per slice is for an experienced human operator to run a series of test slices. The material removal is then measured using the average focal height of the microscope used for the image montage. Based on the calculated material removal and their experience, the human operator adjusts the recipe, typically by adjusting the polishing times for each pad. This process is repeated with another run of test slices until the target removal amount is achieved. This process is not ideal for several reasons. First, it requires significant human intervention from a skilled operator whose valuable experience could be better used for other pursuits. Second, the calibration of the recipe can require multiple test runs, removing a large amount of material. This is inappropriate for small material-samples since a significant portion of the material-sample will not be sectioned at a consistent rate. An automated MSS controller could both reduce human intervention and improve the performance of the system.

This paper proposes a run-to-run (R2R) controller to reduce operator intervention while improving the precision of material removal. R2R is described as "similar to batch processes but more extensive" [2]. R2R is commonly used to reduce output variance and increase precision in systems that perform repetitive processes [3] [4] [5]. R2R assumes that the output is sparsely sampled which allows for a linear regression model to be used. We will be using R2R in combination with constrained optimization with the goal of reducing the variance of the output of the system awhile achieving the target removal amount faster and more repeatable than the "guess and check" current method being used. One of the main challenges is that we do not have access to an analytic first-principles model that maps recipes to material removal. Instead, we will use a data-driven method to synthesize of run-to-run (R2R) controller from historical operational data. We will also compare our method with a state of the art

method.

The remainder of this paper is organized as follows, in section II we define the mechanical serial sectioning model, followed by the explicit problem statement. In section III we propose our method and solve the stochastic optimization problem using run-to-run control and present current state of the art methods for comparison. In section IV we present the main results of the paper and compare the results of the purposed method to the state of the art methods. Finally, in section V we address future research and directions for this work.

Notation: The mean and variance of a random variable x will be denoted by $\mathbb{E}[x]$ and $\mathbb{V}[x] = \mathbb{E}[xx^\top] - \mathbb{E}[x]\mathbb{E}[x]^\top$. The Kronecker product of matrices A and B is denoted by $A \otimes B$. The vectorization $\mathrm{vec}(A) \in \mathbb{R}^{nm}$ of a matrix $A \in \mathbb{R}^{n \times m}$ is a vector comprised of the concatenated column vectors of A. For a matrix A with linearly independent columns, its left pseudo-inverse if a matrix A^+ such that $A^+A = I$. In particular, the Moore-Penrose pseudo-inverse is $A^+ = (A^\top A)^{-1}A$.

II. Automating MSS Problem

In this section, we begin with a brief description of the MSS plant, followed by the operational constraints and the control objectives. We then provide a description of the historical data available to us, used for data-driven controller synthesis.

A. The MSS Plant

We begin by considering a deterministic system model for the MSS plant. Due to heuristics of MSS systems we do not have direct access to in-situ measurements, nor can we change the recipe during a run. Thus, we model the MSS system plant as a discrete time algebraic map of the recipe u_i to removal rate y_i

$$y_i = f(u_i, d_i) \tag{1}$$

where $u_i \in \mathbb{R}^{n_u}$ and $y_i \in \mathbb{R}^1$ are the MSS i^{th} system inputs and outputs respectively, and the i^{th} 'time'-index represents the *slice number* for the plant, depicted in Fig. 1a.

We note that the plant (1) 'dynamics' do not depend on the system 'state' y_i, thus representing a static non-linearity mapping $f(u_i, d_i) : \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^1$ that maps recipes u_i and additional hidden variables d_i to the removal rate y_i. The hidden variables $d_i \in \mathbb{R}^{n_d}$ characterize all unknown factors within the system, such as material hardness, thermal effects, sensor measurement error, cutting and polishing pad wear, cavities in the material, etc. Since we do not have access to the hidden variables d_i, it is not possible to learn a model of the static nonlinearity from historical data. Furthermore, even if the plant model were known, we cannot utilize the hidden variables d_i since they are unmeasured and time-varying. Instead, we will use real-time feedback to adjust the recipe u_i online to achieve the desired removal rate $y_i \to r$ where $r \in \mathbb{R}^1$ is the target removal amount, represented in Fig. 1b. Feedback control is ideally suited to the problem of rejecting unknown and varying disturbances d_i. However, the closed-loop system shown in Fig. 1b) will be dynamic due to the controller. We consider the plant (1) to be deterministic, but unknown.

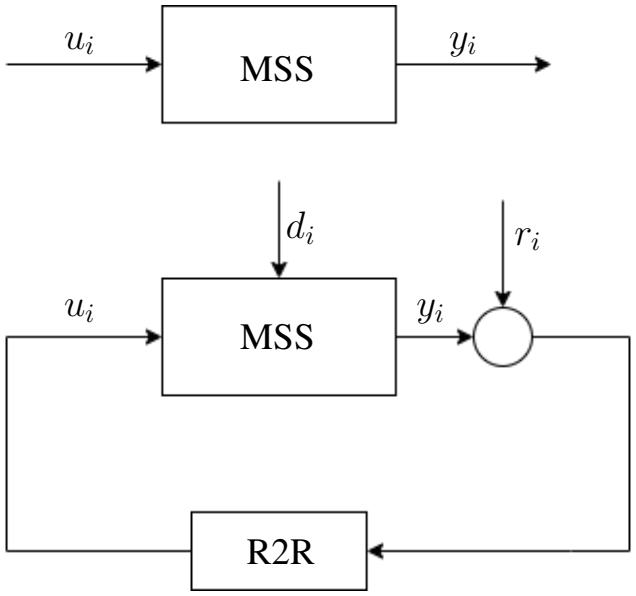

Fig. 1. a.) Depicts the MSS System plant with user full user intervention input ui_i. b.) Depicts the Feedback controllable system block diagram, where the plant is based solely off of plant observations of the MSS and the feed-back R2R controller is defined in section III.

The MSS plant (1) is an over-actuated system; there are multiple $n_u > n_y = 1$ inputs $u_i \in \mathbb{R}^{n_u}$ that can be manipulated to track $y_i \to r$ one output, the desired removal rate $r \in \mathbb{R}^1$. The manipulated inputs are the recipe parameters summarized in Table I. For the automated system, we envision that the MSS will operate with a fixed sequence of multiple cutting and polishing pad types, as well as different solution types for each pad. The automated system will then select the polishing speed and polishing time for each of the pads in the sequence. The vector $u_i \in \mathbb{R}^{n_n}$ of polishing-speeds and polishing-times is the control input (recipe) for the MSS system. For this preliminary work, we will only manipulate the polishing times for a sequence of 2 pads. Importantly for controller development, the removal rate (1) is monotonically non-decreasing with respect to the polish times i.e. polishing for a longer time will not result in less material removed.

The over-actuation of the MSS renders human-in-the-loop operation difficult, requiring the operator to have significant experience and expertise to choose the appropriate recipe $u \in \mathbb{R}^{n_u}$. An automated system will improve the user-friendliness of the MSS. Furthermore, an automated system could harness this over-actuation to improving system performance, shorten the calibration-phase, and reduce human error. Performance objectives will be described in Section II-C.

B. Operational Constraints

Our controller must produce recipes u that are physically implementable by the MSS. The operational constraints

TABLE I
INPUTS (RECIPE PARAMETERS) FOR THE MSS

Input Variable	Constraints
u_{rt} cutting pad type	$1 \leq \gamma \leq 8$
u_{pt} polishing pad type	$1 \leq \gamma \leq 8$
u_s polishing solution type	$1 \leq \gamma \leq 4$
u_ω polishing speed (RPM) for each pad	$1 \leq \gamma \leq 300$
u polishing time for each pad	$5 \leq \gamma$

shown in Table I describe the physical limitations of the MSS system. The maximum number of pads the system is able to hold at a time is eight, therefore $u_{rt} + u_{pt} \leq 8$ for any recipe. For polishing solutions u_s we are able to select one one of the following particle sizes $1\mu m$, $3\mu m$, $6\mu m$, or $9\mu m$ which can be used in any combination with the pad types. The speed at which the pads can be rotated u_ω is limited to 300 RPM. The minimum amount of time a pad can polish for is five seconds, due to ensuring a favorable imaging surface while avoiding over use of polishing pads, constraining polishing pads to more than sixty seconds and less than 200 seconds will be imposed. The constraints for our system is a n_u-dimensional polytope

$$\mathcal{U} = \big\{u \in \mathbb{R}^{n_u} : Hu \leq h\big\}. \tag{2}$$

We will analyze our R2R controller with and without these constraints (2).

C. Control Objectives

Beyond automating the operation of the MSS, our control objectives include improving is performance. Our control objectives can be summarized by the following conceptual stochastic optimization problem

$$\begin{aligned} u_i = \ \arg\min_{u} \quad & \mathbb{V}[y - r|u] && \text{(3a)} \\ \text{s.t.} \quad & \mathbb{E}[y|u] = r && \text{(3b)} \\ & u \in \mathcal{U}. && \text{(3c)} \end{aligned}$$

The desired controller should compute a recipe u such that the expected material removal $\mathbb{E}[y|u]$ matches the target removal amount r i.e. the recipe u should satisfy the equality constraint (3b). Since the MSS is over-actuated $n_u > 1$, there are potentially an infinite-number of recipes $u \in \mathbb{R}^{n_u}$ that can achieve the desired removal (3b). Among these recipes u, we would like to select the recipe that produces the lowest variance (3a) so that the slices have uniform thickness. Finally, the recipe u must be implementable (3c) given the input constraints described in Section II-B.

In Section III, we will translate the conceptual stochastic optimization problem (3) into an implementable deterministic optimization problem. We will use a data-driven approach to formulate this deterministic optimization problem from historical operation data. This deterministic optimization problem will be embedded in a R2R control framework to iteratively ensure that the removal rate converges $y_i \to r$ to the target removal rate r.

D. Historical Operational-Data

We will use a data-driven approach to translate the conceptual stochastic optimization problem (3) into an implementable deterministic form. The available data consists of pairs of recipes $u_i \in \mathbb{R}^{n_u}$ and the resulting removal rate $y_i \in \mathbb{R}^1$. The entire available data set $\mathcal{D} = \{\mathcal{Y}, \mathcal{U}\}$. The data set $\mathcal{D}$ is comprised of historical usage of the plant providing a set of outputs based off of previous operator defined inputs. The set $\mathcal{D} \in \mathbb{R}^{N \times n_\mathcal{D}}$ is the set of N states corresponding to the set of inputs $\mathcal{U} \in \mathbb{R}^{N-1 \times n_\mathcal{U}}$ with respective outputs $\mathcal{Y} \in \mathbb{R}^{N \times n_\mathcal{Y}}$. Using this historical data, we are able to approximate a deterministic model for 1.

III. OPTIMAL RUN-TO-RUN CONTROLLER

In this section, we describe the R2R controller for automating the MSS system. R2R control is the appropriate paradigm for this problem due to the lack of in-situ measurements and our inability to alter the recipe during a slice. Our algorithm embeds a deterministic formulation of the stochastic optimization problem (3) into a R2R framework in order to compute optimal recipes $u^\star_{i+1}$. The R2R framework provides a feedback mechanism for adjusting the recipe $u^\star_{i+1}$ based on the material removal y_i, which is measured after each slice. This feedback is used to reject the hidden variables d_i, which we consider disturbances.

A. Optimal Recipe

The main challenge for MSS controller synthesis is that the plant model (1) mapping recipes u_i to removal rate y_i is an *unknown* static non-linearity. However, since the removal rate (1) is monotonic, we can use a linear approximation

$$y_i = c + b^\top u_i \tag{4}$$

where $c \approx f(u_i, d_i)$ is the drift coefficient and $b \approx \nabla f(u_i, d_i)$ is the slope coefficient around the operating point u_i. The parameters $c \in \mathbb{R}^1$ and $b \in \mathbb{R}^{n_u}$ are uncertain and time-varying due to both the changing linearization point u_i and the hidden variables d_i. To capture this uncertainty, we will models these parameters as stochastic. Although the probability density function of these stochastic variables is unknown, we will use historical-data to quantify our uncertainty using their empirical moments. We will use these empirical moments to translate the conceptual stochastic optimization problem into a deterministic optimization problem. Since we consider stochastic *parameters*, estimating the parameters is non-trivial. We will describe a method for the estimating the mean and variance of c and b in Section II.

First, we translate the stochastic equality constraint (3b) into a deterministic constraint based on empirical moments. Substituting the stochastic linear model (4) into the equality constraint (3b) yields M

$$\mathbb{E}[y_i|u_i] = \mathbb{E}[c + b^\top u_i|u_i] = r_i.$$

Exploiting the linearity of the expectation, we obtain

$$\mathbb{E}[c] + \mathbb{E}[b]^\top u_i = r_i$$

where r_i and u_i are deterministic. This equality becomes the deterministic constraints (6b) when the expectations $\mathbb{E}[c]$ and $\mathbb{E}[b]$ are replaced by their empirical estimates $\mu_c \approx \mathbb{E}[c]$ and $\mu_b \approx \mathbb{E}[b]$.

Next, we translate the conceptual stochastic cost (3a) into deterministic cost function. Substituting the stochastic linear model (4) into the cost (3a) yields

$$\mathbb{V}[y_i|u_i] = \mathbb{E}[(y_i - r_i)^2|u_i] = \mathbb{E}[(c + b^\top u_i - r_i)^2|u_i] \quad (5)$$

where the mean value of y_i is r_i due to the equality constraint (3b). Expanding the cost, yields

$$\mathbb{V}[y_i|u_i] \propto u_i^\top \mathbb{E}[bb^\top] u_i + 2u_i^\top \mathbb{E}[bc] - 2u_i^\top \mathbb{E}[b] r_i$$

where the terms from $\mathbb{E}[(c + r_i)^2]$ were omitted since they do not depend on the decision variables u_i. Substituting the empirical estimates $\mathbb{E}[bb^\top] \approx \Sigma_{bb} + \mu_b \mu_b^\top$ and $\mathbb{E}[bc] \approx \Sigma_{bc} + \mu_b \mu_c$, we obtain

$$\mathbb{V}[y_i|u_i] \propto u_i^\top (\Sigma_{bb} + \mu_b \mu_b^\top) u_i + 2u_i^\top \Sigma_{bc} - 2u_i^\top \mu_b (\mu_c - r_i)$$

Since $\mu_c - r_i = -\mu_b^\top u_i$ according to (6b), we obtain

$$\mathbb{V}[y_i|u_i] \propto u_i^\top (\Sigma_{bb} - \mu_b \mu_b^\top) u_i + 2u_i^\top \Sigma_{bc}$$

Finally, noting that $\mu_b^\top u_i$ is constant, we obtain the deterministic cost (6a). Thus, the conceptual stochastic optimization problem (3) can now be approximated by the following deterministic optimization problem

$$\begin{aligned} u_i = \ & \arg\min_u \quad u^\top \Sigma_{bb} u + 2u^\top \Sigma_{bc} && \text{(6a)} \\ & \text{s.t.} \quad \mu_c + \mu_b^\top u_i = r_i && \text{(6b)} \\ & \qquad u_i \in \mathcal{U} && \text{(6c)} \end{aligned}$$

where the approximation is due to the use of empirical estimates of the means μ_c,μ_b and variances Σ_{bb},Σ_{bc} of the parameters. Conveniently, the problem formulation (6) only requires second-order statics for the model (4) parameters, which we will estimate in Section III-C. Solving (6) will produce the optimal recipe $u_i^\star$.

B. Run-to-Run Controller

Algorithm 1 Optimal Run-to-Run Control

1: Implement initial recipe u_0
2: **repeat**
3: Measure material remove y_i for i-th slice
4: Update (7) drift coefficient μ_c
5: Solve (6) for optimal recipe u_i
6: Implement recipe u_i
7: **until** All slices complete

The deterministic optimization problem (6) is static. Thus in this section, we embed this optimization problem (6) into an R2R framework to provide feedback. Our R2R algorithm iteratively adjust the recipe $u_{i+1}^\star$ based on the measured material removed y_i during the previous slice i. The R2R feedback allows the material removal to converge $y_i \to r$ to the desired removal rate r while rejecting the unmeasured disturbances d_i.

Our R2R controller is described by Algorithm 1. After each slice, the R2R measures the resulting material removal y_i. The difference $y_i - r$ between the actual y_i and desired r removal rate is used to update the drift coefficient μ_c. We update the drift coefficient using exponentially weighted moving average (EWMA) dynamics [6]

$$\mu_c^+ = \mu_c + \lambda(y_i - r) \quad (7)$$

where μ_c^+ is the updated drift coefficient and $\lambda \in [0, 1]$ is a tuning parameter. The EWMA update-law has many beneficial properties [6]. The optimization problem (6) is solved with the updated drift coefficient μ_c to obtain a new recipe u_i which is then implemented. Thus, we can interpret the drift coefficient μ_c as a state and the equality constraint (6b) as dynamics. The R2R controller continues to refine the recipe u_i until all slices have been completed.

The R2R Algorithm 1 indirectly adjusts the recipe u_i by updating the drift coefficient μ_c.

C. Mean and Covariance Estimation

We use the historical operational data $\{y_i, u_i\}_{i=1}^M$ to empirically estimate the moments $\mu_\theta \approx \mathbb{E}[\theta]$ and $\Sigma_{\theta\theta} - \mu_b \mu_b^\top \approx \mathbb{E}[\theta\theta^\top]$ of the parameters $\theta = (c, b)$ of (4) used in the optimization problem (6). This is non-trivial since the parameters $c \approx f(u, d)$ and $b \approx \nabla_u f(u, d)$ are uncertain due to their dependence on the hidden variables d and the changing operating point u. This produces the regressor equation

$$y_i = \mu_c + \mu_b^\top u_i + \sigma_i$$

where the non-determinism is lumped into the 'noise' term $\sigma_i = c - \mu_c + (b - \mu_b)^\top u_i + \bar{\sigma}$. This noise σ_i include typical measurement noise $\bar{\sigma}$, as well as, the 'noise' $c - \mu_c + (b - \mu_b)^\top u_i$ due to parameter uncertainty. Clearly the combined noise σ_i is *not* independent identically distributed (IID); it is not identically distributed nor is it independently distributed from the data $\{y_i, u_i\}_{i=1}^M$. Thus, least-squares estimation is not applicable. Instead, we use the generalized method of moments (GMM) to estimate the moments μ_θ and $\Sigma_{\theta\theta}$ of the parameters since it is applicable to non-IID noise.

GMM poses the moment estimation problem as an optimization problem

$$\hat{\theta} = \arg\min_\theta \left\| \frac{1}{N} \sum_{i=1}^N g(z_i, \theta) \right\|^2 \quad (8)$$

where $z_i = (y_i, u_i)$ is the i-th data point. The sum in (8) is the empirical approximation

$$\mathbb{E}\left[g(y_i, \theta)\right] \approx \frac{1}{N} \sum_{i=1}^N g(z_i, \theta) \quad (9)$$

of a vector-valued function $\mathbb{E}\left[g(y_i, \theta)\right]$ that is zero if and only if the estimated parameter θ matches $\theta = \bar{\theta}$ the true parameter $\bar{\theta}$.

For estimating the mean μ_θ of the parameters $\theta = (c, b)$, we use the vector-valued function $\mathbb{E}\big[\phi_i(y_i - \phi_i^\top \mu_\theta)\big]$. For

this function, it is known [7] that the GMM estimator (8) is a weighted least-squares estimator

$$\hat{\mu}_\theta = \left(\sum\nolimits_{i=1}^N \phi_i \phi_i^\top\right)^{-1} \left(\sum\nolimits_{i=1}^N \phi_i y_i\right) \tag{10}$$

where $\phi_i = (1, u_i)$. Estimating the covariance matrix $\Sigma_{\theta\theta}$ of the parameters $\theta = (a, b)$ is more nuanced and summarized by the following proposition.

Proposition 1. *Let* $\operatorname{rank}\left(\sum_{i=1}^N \phi_i\phi_i^\top \otimes \phi_i\phi_i^\top\right) = \frac{n^2+n}{2}$. *Then*

$$\operatorname{vec}(\hat{\Sigma}) = \left(\sum_{i=1}^N \phi_i\phi_i^\top \otimes \phi_i\phi_i^\top\right)^\dagger \left(\sum_{i=1}^N \epsilon_i^2 \phi_i \otimes \phi_i\right) \tag{11}$$

is a consistent estimator of the covariance matrix where $\epsilon_i = y_i - \phi_i^\top \hat{\mu}_\theta$ *is the estimation error using the mean* $\hat{\mu}_\theta$ *obtained from* (10)*.*

Proof. The covariance estimate (11) is obtained using the vector-valued function $\mathbb{E}\left[\operatorname{vec}(\phi_i\phi_i^\top)(\epsilon_i^2 - \phi_i^\top\Sigma\phi_i)\right]$. Using the linearity of expectation, it can be verified that this function is zero if and only if $\Sigma \succ 0$ is the true covariance

$$\begin{aligned}&\mathbb{E}\left[\operatorname{vec}(\phi_i\phi_i^\top)(\epsilon_i^2 - \phi_i^\top\Sigma\phi_i)\right]\\ &\quad= \operatorname{vec}(\phi_i\phi_i^\top)\phi_i^\top\big(\mathbb{E}\left[(\theta-\mu_\theta)(\theta-\mu_\theta)^\top\right] - \Sigma\big)\phi_i = 0\end{aligned}$$

where $y_i = \phi_i^\top\theta$ and $\Sigma = \bar{\Sigma}_{\theta\theta}$ is the only *positive definite* matrix for which this expression is zero.

The resulting GMM problem (8) has the form

$$\operatorname{vec}(\hat{\Sigma}) = \arg\min_{\Sigma\succ 0} \left\| \frac{1}{N}\sum_{i=1}^N \operatorname{vec}(\phi_i\phi_i^\top)(\epsilon_i^2 - \phi_i^\top\Sigma\phi_i)\right\|^2 \tag{12}$$

Since $\phi_i^\top\Sigma\phi_i = (\phi_i\otimes\phi_i)^\top\operatorname{vec}(\Sigma)$ and $\operatorname{vec}(\phi_i\phi_i^\top) = \phi_i\otimes\phi_i$, we obtain the first-order optimality conditions

$$S_{\phi\phi}\big(S_{\phi\epsilon} - S_{\phi\phi}\operatorname{vec}(\Sigma)\big) = 0 \tag{13}$$

where $S_{\phi\phi} = \sum_{i=1}^N \phi_i\phi_i^\top\otimes\phi_i\phi_i^\top$ and $S_{\phi\epsilon} = \sum_{i=1}^N \epsilon_i^2\phi_i\otimes\phi_i$. The crucial observation is that the matrix $S_{\phi\phi}$ is only positive *semi*-definite. Therefore, the non-strictly-convex quadratic program (12) has an infinite number of solutions of the form

$$\operatorname{vec}(\hat{\Sigma}) = S_{\phi\phi}^+ S_{\phi\epsilon} + N \tag{14}$$

where $S_{\phi\phi}^+$ is the Moore-Penrose pseudo-inverse of $S_{\phi\phi}$ and $N \in \operatorname{null}(S_{\phi\phi})$. We will show that only one of these solutions corresponds to a positive-definite covariance estimate $\hat{\Sigma}$, specifically $N = 0$.

First, we will show that the null-space $\operatorname{null}(S_{\phi\phi}) \subseteq \mathbb{R}^{n^2}$ is the set of vectorizations $\operatorname{vec}(M)$ of skew-symmetric matrices $M = -M^\top \in \mathbb{R}^{n\times n}$. The vector-space of skew-symmetric matrices has a basis $e_ie_j^\top - e_je_i^\top$ for $j > i \in \{1,\dots,n\}$ where $e_i \in \mathbb{R}^n$ is the i-th standard basis vector. By properties of Kronecker products, we have

$$\begin{aligned}&S_{\phi\phi}\operatorname{vec}(e_ie_j^\top - e_je_i^\top)\\ &\quad= \sum\nolimits_{i=1}^N \big(\phi_i\phi_i^\top\otimes\phi_i\phi_i^\top\big)\big(e_i\otimes e_j - e_j\otimes e_i\big)\\ &\quad= \sum\nolimits_{i=1}^N \big([\phi_i]_i[\phi_i]_j - [\phi_i]_i[\phi_i]_j\big)\phi_i\otimes\phi_i = 0\end{aligned}$$

where $\operatorname{vec}(e_ie_j^\top - e_je_i^\top) = \operatorname{vec}(e_ie_j^\top) - \operatorname{vec}(e_je_i^\top) = e_i\otimes e_j - e_j\otimes e_i$. Thus, $\operatorname{vec}(\operatorname{skew}(\mathbb{R}^{n\times n})) \subseteq \operatorname{null}(S_{\phi\phi})$. Since $\operatorname{rank}(S_{\phi\phi}) = (n^2+n)/2$ and $\operatorname{vec}(\operatorname{skew}(\mathbb{R}^{n\times n}))$ has dimension $(n^2-n)/2$, we can conclude $\operatorname{vec}(\operatorname{skew}(\mathbb{R}^{n\times n})) = \operatorname{null}(S_{\phi\phi})$.

Since $\phi_i\otimes\phi_i = \operatorname{vec}(\phi_i\phi_i^\top)$, we have $S_{\phi\epsilon} \in \operatorname{vec}(\operatorname{sym}(\mathbb{R}^{n\times n}))$ is the vectorization of a symmetric matrix. Furthermore, since $\operatorname{null}(S_{\phi\phi}) = \operatorname{vec}(\operatorname{skew}(\mathbb{R}^{n\times n}))$, the Moore-Penrose pseudo-inverse $S_{\phi\phi}^+$ will map vectorization of a symmetric matrices to vectorization of a symmetric matrices. Thus, $S_{\phi\phi}^+S_{\phi\epsilon}$ is the vectorization of a symmetric matrix. Finally, we note that adding the vectorization of a non-symmetric matrix $N \in \operatorname{null}(S_{\phi\phi})$ in (14) will break symmetry. Thus, optimal solution of (12) is $S_{\phi\phi}^+S_{\phi\epsilon}$ i.e. (11). The consistency of the estimator follows from properties of GMM [7]. □

D. Comparison with Existing R2R Controllers

In this section, we compare our R2R controller in Algorithm 1 with existing R2R controllers from the literature. As noted in the survey [2], most R2R controllers have the following integral dynamics

$$u_{i+1} = u_i + \lambda\mu_b^+(r - y_i) \tag{15}$$

where $\mu_b^+ = \mu_b/\mu_b^\top\mu_b$ is the Moore-Penrose pseudo-inverse and $\lambda \in [0,1]$ is a tuning parameter. See equation (16) from [2] for details. We will show that our R2R controller has integral dynamics (15) when the input constraints (6c) are ignore, although with a novel pseudo-inverse. In contrast, when the input constraints are ignore the integral dynamics no longer apply.

Without the input constraints (6c), the Karush-Kuhn-Tucker (KKT) optimality conditions are

$$\begin{bmatrix} 2\Sigma_{bb} & \mu_b \\ \mu_b^\top & 0\end{bmatrix}\begin{bmatrix} u \\ \nu\end{bmatrix} = \begin{bmatrix} -2\Sigma_{bc} \\ r - \mu_c\end{bmatrix} \tag{16}$$

where $\nu \in \mathbb{R}$ is the dual variable associated with the equality constraint (6b). Solving (16) for u, we obtain the control-law

$$u = \mu_b^\dagger(r - \mu_c) + \tfrac{1}{2}\big(I - \mu_b^\dagger\mu_b^\top\big)\Sigma_{bb}^{-1}\Sigma_{bc} \tag{17}$$

where $\mu_b^\dagger = \Sigma_{bb}^{-1}\mu_b/(\mu_b^\top\Sigma_{bb}^{-1}\mu_b)$ is an alternative pseudo-inverse of μ_b i.e.

$$\mu_b^\top\mu_b^\dagger = \mu_b^\top\Sigma_{bb}^{-1}\mu_b/(\mu_b^\top\Sigma_{bb}^{-1}\mu_b) = 1.$$

This pseudo-inverse was derived from (6) to minimizes the variance $\mathbb{E}[(y-r)^2|u]$ of the material removal. Combining (17) with the EWMA dynamics (7), we obtain the following integral dynamics

$$u_{i+1} = u_i + \lambda\mu_b^\dagger(r - y_i) \tag{18}$$

with the specific initial condition

$$u_0 = \mu_b^\dagger(r-\mu_{c0}) + \tfrac{1}{2}\big(I-\mu_b^\dagger\mu_b^\top\big)\Sigma_{bb}^{-1}\Sigma_{bc}.$$

This initial condition is important since otherwise the dynamic controller (18) would not include the second-term which compensates for possible cross-correlation between the model (4) parameters c and b. Note that our integral dynamics (18) match the literature dynamics (15).

Next, we show our R2R Algorithm 1 does not necessarily have literature dynamics (15) when the input constraints (6c) are included. With the input constraints (6c), the KKT optimality conditions for (6) are

$$\begin{bmatrix} 2\Sigma_{bb} & \mu_b & H_{\mathcal{A}}^\top \\ \mu_b^\top & 0 & 0 \\ H_{\mathcal{A}} & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \nu \\ \lambda_{\mathcal{A}} \end{bmatrix} = \begin{bmatrix} -2\Sigma_{bc} \\ r-\mu_c \\ h_{\mathcal{A}} \end{bmatrix} \tag{19}$$

where H and h are the half-space parameters of the input constraints (2) and $H_{\mathcal{A}}$ and $h_{\mathcal{A}}$ are the rows corresponding to the subset $\mathcal{A}$ of constraints that are active at the optimal. The active dual variables are denoted by $\lambda_{\mathcal{A}} \geq 0$ where the dual variable corresponding to inactive constraints are zero. Through brute-force computation, we obtain

$$u = \mu_b^\ddagger(r-\mu_c) + \tfrac{1}{2}\big(I-\mu_b^\ddagger\mu_b^\top\big)(\Sigma_{bb}^{-1}\Sigma_{bc} + \Gamma h_{\mathcal{A}}) \tag{20}$$

where $\Gamma = \Sigma_{bb}^{-1}H_{\mathcal{A}}^\top(H_{\mathcal{A}}\Sigma_{bb}^{-1}H_{\mathcal{A}}^\top)^{-1}$ and $\mu_b^\ddagger$ is yet another pseudo-inverse of μ_b given by

$$\mu_b^\ddagger = \frac{(\Sigma_{bb}^{-1} - \Sigma_{bb}^{-1}H_{\mathcal{A}}^\top(H_{\mathcal{A}}\Sigma_{bb}^{-1}H_{\mathcal{A}}^\top)^{-1}H_{\mathcal{A}}\Sigma_{bb}^{-1})\mu_b}{\mu_b^\top(\Sigma_{bb}^{-1} - \Sigma_{bb}^{-1}H_{\mathcal{A}}^\top(H_{\mathcal{A}}\Sigma_{bb}^{-1}H_{\mathcal{A}}^\top)^{-1}H_{\mathcal{A}}\Sigma_{bb}^{-1})\mu_b}.$$

Note that if (6) is feasible then μ_b does not lie in the null-space of $\Sigma_{bb}^{-1} - \Sigma_{bb}^{-1}H_{\mathcal{A}}^\top(H_{\mathcal{A}}\Sigma_{bb}^{-1}H_{\mathcal{A}}^\top)^{-1}H_{\mathcal{A}}\Sigma_{bb}^{-1}$. This follows from the fact that the active inequality constraints cannot bind the equality constraint (6b).

Although (20) has a similar structure as (17), it cannot necessarily be transformed into the integral-form (15). As μ_c changes (7), the optimal active-set $\mathcal{A}$ can change. Thus, the pseudo-inverse $\mu_b^\ddagger$ and matrix Γ are time-varying. Thus, the nonlinear map provide by the optimization problem (6) replaces rather than integrates (15) the control inputs. Note that, although our R2R controller Algorithm 1 does not have the integral dynamics (15), it is still dynamic due to the EWMA dynamics (7).

Finally note that our R2R controller can be trivially put in the general form $u_{i+1} = \alpha u_i + \delta u_i$ given by equation (17) in [2] since any arbitrary feedback controller $\kappa(x)$ can be written in this form by defining $\delta u_i = \kappa(x) - \alpha u_i$.

IV. Numerical Results

In this section, we present numerical results demonstrating our R2R controller. First, we present results verifying our stochastic model (4) of the material removal (1). Next, we present simulation results that demonstrate our R2R controller for an unknown static nonlinearity (1). Finally, we present simulations results our R2R controller for the stochastic model (4) where the parameters c and b are time-varying with a Gaussian distribution.

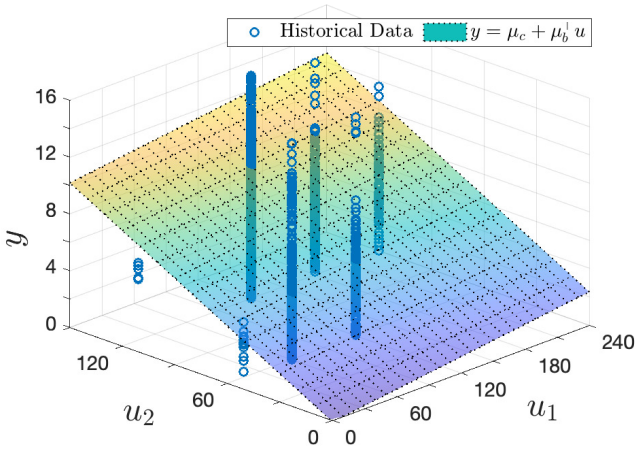

Fig. 2. Fit of stochastic linear model (4) using historical data.

Throughout this section, we consider the MSS with 2 pads to allow us to plot $y_i \in \mathbb{R}^1$ versus $u_i \in \mathbb{R}^2$.

A. Parameter Estimation and Model Validation

We use historical operational data to estimate the mean and variance of the stochastic linear model (4). The historical data was mined from multiple data files, which contain the states of the system, inputs (polish times, RPM, pads used) and outputs (microscope focus height). We utilize the observations from the system plant which come in the form of multiple data files, which contain the inputs (polish times, RPM, pads used) and outputs (microscope focus height). We developed an script that extracts this data from thousands of separate text-files and collects the data into a unified data set $\mathcal{D} = (\mathcal{Y},\mathcal{U})$ where $\mathcal{U} \in \mathbb{R}^{n\times 2}$ are the recipes and $\mathcal{Y} \in \mathbb{R}^{n\times 1}$ are the resulting amounts of material removed. When a slice is imaged, the focal height of the microscope is recorded for each imagine in the montage. The average focal height is then calculated and used to estimate the amount of material removed for each slice. Microscope auto focus errors can occur which will cause incorrect average focal height readings which can lead to readings of negative or minimal removal amounts. Therefore, the need to pre-process the data by removing outliers is required. This includes removing all data points associated with negative values. Also, we compute a preliminary estimate of the mean and variance of the parameters c and b. Any data-points $y_i = c + b^\top u_i$ outside of 3 standard deviations of the estimated mean value $y_i = \mu_c + \mu_b^\top u_i$ are removed.

Once pre-processing is complete, we use the historical data to estimated the mean and variance of the parameters of the stochastic model (4) using the GMM method described in Section III-C. For the mean drift μ_c and mean slope μ_b, the GMM estimator is equivalent to a least-squares estimator. The curve-fit of the mean $y_i = \mu_c + \mu_b^\top u_i$ of (4) is shown in Fig. 2.

Fig. 2 shows that the available historical data is not very exciting (in the sense of persistency of excitation). The expert human operators tend to use a few different recipes and the

polish times are round numbers, typically multiples of 60 seconds. Indeed, the excitation of this input data u_i is

$$\underline{\sigma}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{u_i u_i^\top}{u_i^\top u_i}\right) = 7773.5$$

where $\underline{\sigma}(\cdot)$ is the smallest singular-value of a matrix. While the low-level of excitation makes it difficult to accurately estimate the parameters, it demonstrates the room for improvement through automation. The proposed R2R control Algorithm 1 will not artificially restrict itself to a small number of recipes with round numbers. This greater flexibility can potentially lead to improved performance. In future work, we will consider active-learning/dual-control to produces more exciting data for our data-driven R2R control design.

Fig. 2 shows that material removal y_i is highly variable. Even when the same recipe $u_i = u$ is used, the resulting removal y_i varies greatly. This is partially due to measurement noise, but the hidden parameters d_i play a significant role. Polishing a soft material will remove far more material than polishing a hard material for the same amount of time. Likewise, a fresh pad will remove material more quickly than an old pad. This high variance of the material removal shown in Fig. 2 motivates our decision to model the material removal as stochastic (4). Furthermore, it motivates our objective of finding recipes that minimize the variance in the removal of material.

B. Simulation Results - Linear Model

In this section, we compare our R2R controller in Algorithm 1 with the literature controller (15) using the Moore-Penrose pseudo-inverse $\mu_b^+ = \mu_b/\mu_b^\top\mu_b$ of μ_b. We also compare our controller without the input constraints (6c). As we showing in Section III-D, without (6c) our controller has the form (15) with the pseudo-inverse

$$\mu_b^\dagger = \frac{\Sigma_{bb}^{-1}\mu_b}{\mu_b^\top\Sigma_{bb}^{-1}\mu_b}. \tag{21}$$

For these simulation results, we model the removal function (1) as

$$y_i = \mu_c + \mu_b^\top u_i$$

where y_i is the removal amount, μ_c is estimated variance of the output, μ_b is the estimated variance of the inputs, and u_i is the inputs.

The simulation results are shown in Fig. 3. For each of the 3 R2R algorithms we show the removal rate y_i and the recipe u_i versus slice i. The desired removal rate is $r = 10$ microns.

Each of the R2R algorithms converged to the desired removal rate $y_i \to r$ after 14 slices. This is fast convergence consider an experiment is typically comprised of hundreds of slices. Furthermore a human operator can require up to 40 slices to find an appropriate recipe for a unique sample. However, the two linear R2R controllers (15) with different pseudo-inverses produced different recipes to achieve the desired removal rate. In the next section, we will examine the benefits of the pseudo-inverse (21).

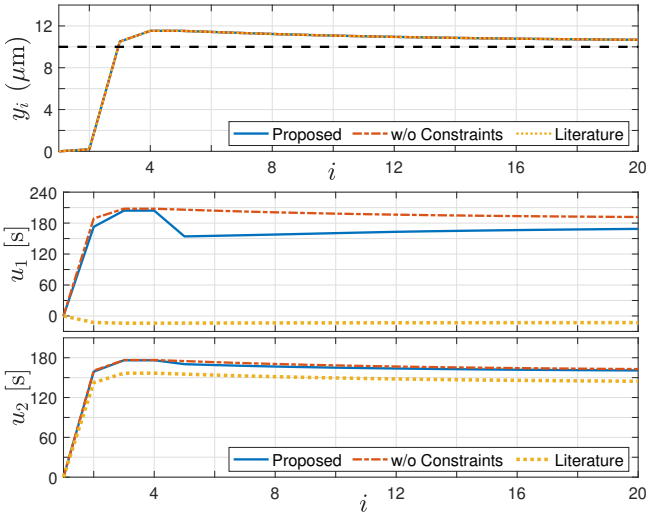

Fig. 3. Simulation results comparing the proposed R2R control with and without constraints with the literature R2R controller using the Moore-Penrose pseudo-inverse.

Both linear R2R controllers (15) produced non implementable recipes. The literature R2R controllers computed negative polishing times for one of the pads. The unconstrained variant of our controller produced excessively long polishing times for one of the pads.

C. Robustness Simulations

In this section, we present simulation results that stress-test the R2R algorithms. We modeled the material removal function (1) using the linear model (4) with time-varying parameters c and b. For each slice i, the parameters were sampled from a Gaussian distribution

$$\begin{bmatrix} c \\ b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_c \\ \mu_b \end{bmatrix}, \begin{bmatrix} \Sigma_{cc} & \Sigma_{cb} \\ \Sigma_{bc} & \Sigma_{bb} \end{bmatrix}\right)$$

where the mean and variance were empirically estimated using the method described in Section III-C. Our R2R controller was compared with the literature controller (15) using the Moore-Penrose pseudo-inverse.

The simulation results are shown in Fig. 4. The target removal rate was $r = 10$ microns. For a fair comparison, the sequence of parameters c_i and b_i was pre-computed so that both algorithms had the same realizations of the random variables.

As Fig. 4 shows, both R2R algorithms keep the average removal rate $\mathbb{E}[y_i]$ around the target removal rate. Indeed, our R2R algorithm had an average removal rate of $\mathbb{E}[y_i] = 9.98$ whereas the literature algorithm had an average $\mathbb{E}[y_i] = 9.99$.

The advantage of our R2R controller is the reduction of the variance of the removal rate $\mathbb{V}[y_i]$. The reduction in the variance is apparent from Fig. 4. Indeed, the variance of the removal rate for our algorithm was $\mathbb{V}[y_i] = 3.33$ whereas the variance of the literature controller was $\mathbb{V}[y_i] = 17.69$. Thus, for these simulation results, our controller reduces the variance of the removal rate by 81%.

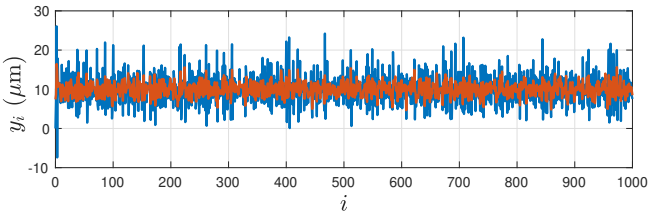

Fig. 4. Comparison of R2R controllers for stochastic removal rates

V. Conclusion and Future Work

We developed a R2R control Algorithm 1 to automate MSS to reduce human intervention. Our R2R controller solves a deterministic formulation (6) of a stochastic optimal control problem (3) that minimizes the variance of the slice thickness. We adopted a data-driven approach to synthesize the R2R controller using historical operational data. We demonstrated our R2R controller through simulation results. We compared our R2R controller with a literature controller. While our R2R controller had similar performance to the state-of-the-art algorithm, the benefit of our approach is that it generated physically implementable recipes. We further demonstrated our R2R controller in stochastic simulations. Our R2R controller reduced the variance by 81%.

Future work entails implementation into the physical to ensuring all real errors were accounted for during simulation and that the algorithm converges the real outputs of the physical system. Improving the current model or creating a more complex model of the system through methods such as Gaussian Process Regression or ”Kriging” to increase the accuracy of stochastic model.

References

[1] J. Madison, J. Spowart, D. Rowenhorst, J. Fiedler, and T. Pollock, “Characterization of three-dimensional dendritic structures in nickel-base single crystals for investigation of defect formation,” in *Proceedings of the Superalloys Conference*. Roger C. Reed and at. Champion Pennsylvania USA The Minerals, Metals …, 2008, pp. 881–888.

[2] Y. Wang, F. Gao, and F. J. Doyle, “Survey on iterative learning control, repetitive control, and run-to-run control,” *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[3] A. Rzepniewski and D. E. Hardt, “Development of general multivariable run-by-run control methods with application to a sheet metal forming process,” *International Journal of Machine Tools & Manufacture*, vol. 48, pp. 599–608, 2008.

[4] R. Ganesan, T. K. Das, and K. M. Ramachandran, “A multiresolution analysis-assisted reinforcement learning approach to run-by-run control,” *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 2, pp. 182–193, 2007.

[5] J. Busch and W. Marquardt, “Run-to-run control of membrane filtration in wastewater treatment - an experimental study 1 1the financial support by the dfg (german research foundation) in the project “optimization-based process control of chemical processes” and the excellent cooperation with koch membrane systems gmbh are gratefully acknowledged.” *IFAC Proceedings Volumes*, vol. 40, no. 5, pp. 195–200, 2007, 8th IFAC Symposium on Dynamics and Control of Process Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015317237

[6] J. Moyne, “Run-to-run control in semiconductor manufacturing,” in *Encyclopedia of Systems and Control*. Springer London, 2014, pp. 1–7.

[7] L. P. Hansen, “Large sample properties of generalized method of moments estimators,” *Econometrica*, vol. 50, no. 4, pp. 1029–1054, 1982.