2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A Framework for Simultaneous Workpiece Registration in Robotic Machining Applications

Steffan Lloyd[1], Rishad Irani[1], and Mojtaba Ahmadi[1]

***Abstract*— This article presents a novel framework called *Simultaneous Registration and Machining* (SRAM), a generalized method to improve workpiece registration using real-time acquired data in robotic contouring applications. The method allows for online corrections to the toolpath, while a live covariance estimate is simultaneously leveraged to adaptively tune the force controller aggressively when uncertainty is high, but conservatively otherwise to minimize chatter and instability. The SRAM framework is validated in simulation and shown to significantly reduce the path corrections required from the force controller, while correctly predicting optimal controller tuning adaptations. The SRAM method is proposed to improve force control stability, increase peripheral accuracy, smooth surface finish, and reduce cycle times in contouring applications.**

I. Introduction

Current industrial robot usage is primarily limited to simple, non-interactive tasks. Interactive machining operations, such as deburring, grinding, milling, or polishing, are currently only implemented on industrial robots in few specific situations and account for only 0.6% of global usage [1]. Many reasons are given for this low adoption, but the most cited are poor robot accuracy, difficulty registering workpieces in the workspace, and slow cycle times due to high process complexity [2]–[5]. Instead, manual methods remain prevalent in many industries, with significant associated costs [6], [7]. Thus, it is desirable to develop novel methods of simplifying robotic machining and increasing robot accuracy.

Consider Fig. 1, which illustrates a serial manipulator in a typical machining application. The positional error between the tool and the workpiece comes from two sources: the accuracy of the manipulator (between the world and tool frames) and the workpiece registration (between the world and part frames). The former can be improved through well-documented, albeit complex, kinematic calibration procedures [8], [9]. However, the latter is difficult to correct as localization must happen on the fly for each new part. Improved workpiece registration is associated with increased cost or increased cycle times. As such, a trade-off exists between process accuracy, cost, and efficiency.

Various techniques have been researched and proposed as a solution to this problem. Precise jigging can reduce workpiece registration errors [10]; however, it is only applicable in applications with low workpiece variability, has a high associated cost, and cannot equally correct for errors in manipulator accuracy. Laser scanning can localize parts accurately and correct for both registration and manipulator accuracy [11]–[17]. However, they are costly, and slow since the scanner must be physically moved over the part to build a point cloud for processing. Moreover, the reflective surface of many metal workpieces can lead to incorrect data, in particular near workpiece imperfections. Touch-off localization [10], [18]–[20] can give accurate results but is exceedingly slow since robot speed is limited near an unknown object, and as the number of touch-offs is small, localization data is sparse and sensitive to outliers caused by imperfections such as burrs, flashing or scallops.

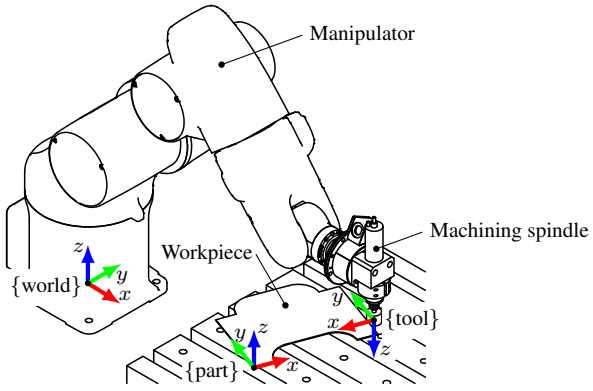


Fig. 1. A robotic manipulator performing a force-controlled edge-processing operation on a sheet metal part, supported by a vacuum table.

Vision systems are attractive as they are fast and inexpensive [2], and have been used in stereo and monocular setups to localize workpieces [21]–[25]. Unfortunately, vision systems provide limited accuracy [26], [27], which is insufficient for most machining operations [11]. Moreover, their results can be skewed by workpiece imperfections, and additional issues arise in the image processing and object detection steps, reducing accuracy and reliability.

When sufficiently accurate part registration is not possible, force control is often used [20], [28], [29], typically implemented as an admittance controller [30], which defines a programmed dynamic relationship between the interaction force and the position of the tool—effectively placing a "virtual suspension system" between the workpiece and the tool and allowing for path errors to be corrected. However, admittance control also has shortcomings. The controller can be tuned to have low damping and behave similarly to a pure force controller with ideal edge tracking. However, in practice, limited robot bandwidth causes stability issues

[1] Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mails: {Steffan.Lloyd, Rishad.Irani, Mojtaba.Ahmadi}@carleton.ca).

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) grants RGPIN-2022-04934, RGPIN-2021-04207, and CRDPJ 514258-17, and industrial partner Arnprior Aerospace Inc. Support for Steffan Lloyd was provided by NSERC CGS-D.

(chatter) with this low-damped configuration, particularly if the contact is rigid [31]. Conversely, the controller may be tuned to have high damping and behave similarly to a pure position (PID) controller. Unfortunately, this highly damped setup loses the ability to quickly compensate for path errors, and operations must be slowed accordingly. In machining applications, the above issues are compounded further. Tolerances are tight relative to robot accuracy [13], [14], and the contact is rigid, forcing a damped tuning to avoid oscillations. Simultaneously, large, high-frequency disturbances are imparted into the robot from the machining spindle, rotating at high speed and further destabilizing the controller. Such oscillations lead to poor surface finish, premature tool wear, and inaccurate peripheral tolerances. Additionally, tool wear causes changes in the tool radius and leads to additional path error [26], [32]. Finally, admittance control allows for edge-following; however, in many machining operations, defects such as burrs, scallops, or flashing must be removed from the workpiece edge. The controller cannot distinguish between these imperfections and the true part edge, so defects may be smoothed but not entirely removed.

Nominally, an admittance controller is reactive, and responds to its environment—but does not learn from it. Consider Fig. 2, which shows the admittance path around a workpiece with registration error. As the tool first approaches the left side of the part, the controller corrects for the horizontal error, then behaves nominally. At the first top-left corner, the controller changes its adaptation to the vertical error, which is unrelated to the horizontal error. However, the controller must repeat this adaptation on the subsequent top-right and bottom-right corners—despite having previously observed these errors. These unnecessary adaptations result in poorer edge tracking and degraded surface finish quality; however, without knowledge of the workpiece geometry, the controller cannot anticipate the relationships between errors, and behavior is poor near corners and other peripheral complexities. Unfortunately, little work has been done to integrate knowledge obtained *during the machining process* into real-time operations, despite the apparent benefits.

The current work seeks to address this knowledge gap through *Simultaneous Registration and Machining* (SRAM), a novel control framework for admittance-controlled contouring applications. SRAM uses contact force and positional data to perform online workpiece registration correction and identification of tool size. Moreover, SRAM optimally adapts the admittance tuning to current process requirements and conditions, providing many benefits over typical implementations. Initial registration can be less precise, so less accurate, but faster and cheaper methods such as vision systems can be used. Cycle times can be decreased since faster registration methods are used, and feedrates can be increased since the admittance parameters are tuned in real-time to optimally maintain contact with the workpiece. Adaptive admittance tuning can improve controller stability and surface finish, as well as ensure that imperfections such as burrs or scallops are fully removed. SRAM is a generalized framework and can be readily extended to many applications to improve reliability and speed.

This article makes several contributions. Firstly, we introduce a novel generalized method for using online force and position data to improve workpiece registration, estimate tool size, and allow the robot path to be corrected in real-time to match the true workpiece position. Second, we develop a method of adaptively tuning admittance controllers based on real-time registration state covariance estimates, allowing for fast admittance corrections when workpiece location information is limited, but smoothly transitioning to slower, more stable behavior when possible. Finally, the efficacy of the localization framework is demonstrated by simulating robot-tool interactions on a sample workpiece and visualizing the real-time correction and controller tuning. The paper is organized as follows. Section II describes the problem statement and notation. Section III describes the SRAM controller architecture and its components. Lastly, Section IV validates the proposed algorithm in a simulated contour-following application.

II. Nomenclature & Problem Summary

We concern ourselves with the robotic machining of planar parts, as shown in Fig. 1. Let $\mathbf{T}_p^*$ denote the homogeneous transformation matrix between the world and part frames, and let $\mathbf{T}_p$ denote a known estimate of this transform, obtained a priori via a vision system, jigging, or similar. The machining tool is modeled as a cylinder of radius r_t^*, and we let r_t denote an a priori estimate of this quantity. The geometry of the part is assumed to be available in terms of primitives such as lines, arcs, and points, and it is assumed

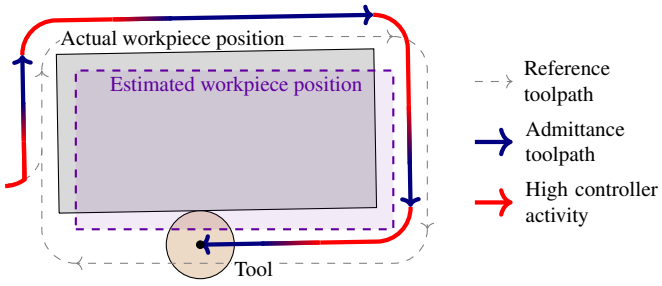


Fig. 2. Admittance controller activity in a contouring application.

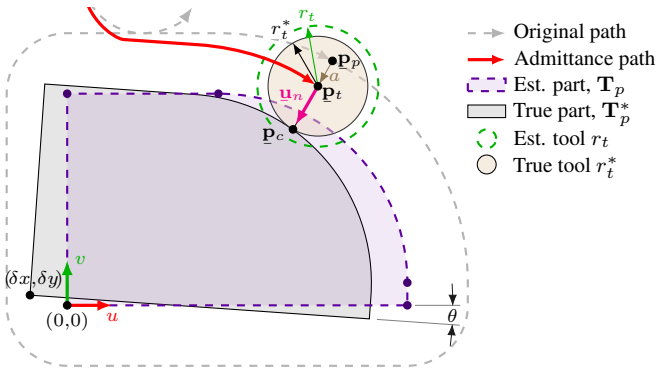


Fig. 3. Visualization of the SRAM setup and nomenclature.

that a contouring path is planned around the part, as shown in Fig. 3, consisting of a time-series of 3D points $\mathbf{p}_p$, and unit-length normal direction vectors $\mathbf{u}_n$. Such path planning is has been addressed in literature [18], [19], [21], [33] and is possible in many CAD/CAM software packages [34], [35]. An admittance controller is employed to correct for path errors by modifying the tool center along the normal axis $\mathbf{u}_n$ by an admittance correction a to maintain a desired normal force f_d from the measured force f, such that

$$\mathbf{p}_t = \mathbf{p}_p + a\,\mathbf{u}_n, \tag{1}$$

$$\ddot{a} = \tfrac{1}{m_a}\big[f_d - f - b_a\,\dot{a} - k_a\,a\big], \quad a = \iint \ddot{a}\;dt\,dt, \tag{2}$$

where m_a, k_a, and b_a are virtual mass-spring-damper coefficients used to tune the "virtual suspension" of the controller.

In the current work, workpieces are assumed planar, and we define the part plane Π_p from the xy plane of $\mathbf{T}_p$ as

$$\Pi_p = \big\{\mathbf{p} \in \mathbb{R}^3 \;\big|\; \mathbf{p} = \mathbf{d}_p + u\,\mathbf{u}_{xp} + v\,\mathbf{u}_{yp},\; u, v \in \mathbb{R}\big\}, \tag{3}$$

where $\mathbf{u}_{xp}$ and $\mathbf{u}_{yp}$ are the unit vectors along the x and y axes of $\mathbf{T}_p$, $\mathbf{d}_p$ the translation vector of $\mathbf{T}_p$, and u, v are the planar coordinates $\underline{\mathbf{p}}$ of the point $\mathbf{p}$—noting that in this paper, the underbar $\underline{\square}$ denotes 2D planar variables from their 3D counterparts. Projection of a point $\mathbf{p} \in \mathbb{R}^3$ to a point $\underline{\mathbf{p}} \in \Pi_p$ and back can be accomplished as

$$\underline{\mathbf{p}} = \begin{bmatrix}\mathbf{u}_{xp} & \mathbf{u}_{yp}\end{bmatrix}^{\mathsf{T}}\big(\mathbf{p} - \mathbf{d}_p\big), \quad \mathbf{p} = \begin{bmatrix}\mathbf{u}_{xp} & \mathbf{u}_{yp}\end{bmatrix}\underline{\mathbf{p}} + \mathbf{d}_p. \tag{4}$$

Vectors are similarly converted without the shift of origin $\mathbf{d}_p$.

Within the plane Π_p, the uncertainty of the workpiece and tool can be described using a four-state vector

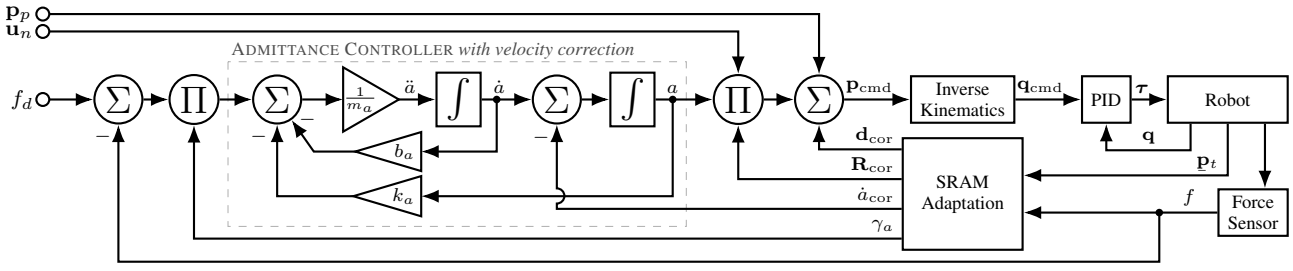
$$\boldsymbol{\rho} = \begin{bmatrix}\delta x & \delta y & \theta & \delta r\end{bmatrix}^{\mathsf{T}}, \tag{5}$$

where δx, δy, and θ are horizontal, vertical, and rotational workpiece corrections, and δr is a tool radius correction. This state vector $\boldsymbol{\rho}$ defines the *corrected* part frame $\tilde{\mathbf{T}}_p$ and tool radius $\tilde{r}_t = r_t + \delta r$—again noting that in this article, the tilde $\tilde{\square}$ denotes the current estimate of a variable, transformed by $\boldsymbol{\rho}$, while those without represent the initial estimate.

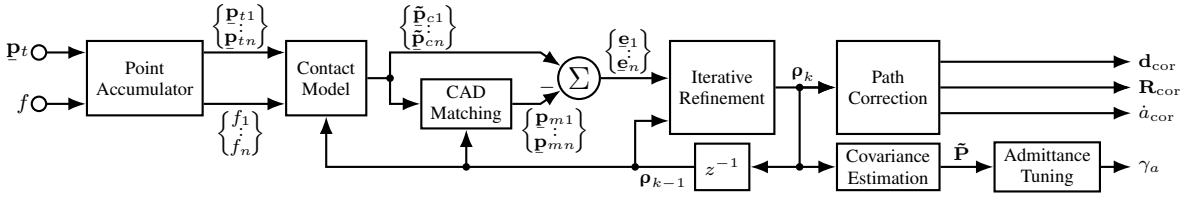
III. Simultaneous Registration & Machining

The SRAM framework modifies the admittance relationship in (2) by allowing for real-time adaptation of the workpiece registration and tool radius to minimize the corrections required by the controller. Simultaneously, a live covariance estimate is computed and leveraged to modulate the admittance tuning optimally. Fig. 4a illustrates the overall SRAM control architecture. A desired force f_d, target pose $\mathbf{p}_p$, and normal vector $\mathbf{u}_n$ are provided by the path planner. A standard admittance controller modulates the target pose along the normal axis to maintain the target force. An inverse kinematics block converts the target pose to joint angles, which are sent to the joint servos. At the same time, a SRAM adaption block records the measured position $\mathbf{p}_t$ and part interaction forces f to update the state estimate $\boldsymbol{\rho}$ and compute a path correction $\mathbf{d}_{\text{cor}}$, a rotational correction matrix $\mathbf{R}_{\text{cor}}$, and an admittance velocity correction term $\dot{a}_{\text{cor}}$. Finally, an admittance tuning parameter γ_a is calculated from an estimate of the state covariance—allowing for the controller to behave rapidly before knowledge of the workpiece location is obtained, then transition naturally to be a stiff controller as the registration improves.

The architecture of the SRAM adaptation system is shown in Fig. 4b. First, the *point accumulator* records the incoming planar tool coordinates $\underline{\mathbf{p}}_t$ and forces f to form a list of all observations in the current run. The *contact model* system estimates workpiece-tool contact points $\underline{\mathbf{p}}_{ci}$ from each observation. The *CAD matching* relates the estimated contact points with corresponding points on the CAD model, $\underline{\mathbf{p}}_{mi}$. The matched points and contact points are subtracted to form a set of residuals $\underline{\mathbf{e}}_i$ and used by the *iterative refinement* block to compute state updates to $\boldsymbol{\rho}$. The *path correction* system computes $\mathbf{d}_{\text{cor}}$, $\mathbf{R}_{\text{cor}}$, and $\dot{a}_{\text{cor}}$ from the state estimate $\boldsymbol{\rho}$. Finally, the *covariance estimation* system estimates a state covariance matrix $\tilde{\mathbf{P}}$, which is then used by the *admittance tuning* block to compute an admittance tuning factor γ_a. These blocks are described further in the following sections.



(a) Architecture of the SRAM algorithm.



(b) Diagram of the SRAM adaptation subsystem.

Fig. 4. Architecture of the proposed SRAM framework, which behaves as an admittance controller but with online correction and tuning terms.

A. Point Accumulator

The point accumulator block's purpose is to build up a list of relevant observations, $\underline{\mathbf{p}}_{ti}$ and f_i, of workpiece-tool contacts. To avoid this list of observations from becoming overly large, we employ two strategies. First, an observation is only considered relevant (and thus logged) if the tool is in contact with the workpiece—this can be deduced by comparing the measured force f_i to a set threshold. Second, observations are logged at a lower frequency than the overall controller by downsampling the input, with appropriate anti-alias lowpass filtering, before processing.

B. Contact Model

The contact model block of the SRAM framework seeks to convert the set of n observed tool coordinates $\underline{\mathbf{p}}_{ti}$ and forces f_i into a set of n workpiece-tool contact points $\underline{\tilde{\mathbf{p}}}_{ci}$. The estimation models the workpiece-tool contact as a spring of stiffness k_t, as shown in Fig. 5. While k_t is shown as the tool stiffness, it in reality combines stiffnesses of the tool, part and manipulator. Accordingly, the block output $\underline{\tilde{\mathbf{p}}}_{ci}$ is

$$\underline{\tilde{\mathbf{p}}}_{ci} = \underline{\mathbf{p}}_{ti} + \left(r_t + \delta r - f_i/k_t\right) \underline{\mathbf{R}}_\theta \, \underline{\mathbf{u}}_{ni}, \tag{6}$$

with $\underline{\mathbf{R}}_\theta$ as the 2D rotation matrix of the angle θ.

C. CAD Matching

The third step in the SRAM method is to match each of the estimated workpiece-tool contact points $\underline{\tilde{\mathbf{p}}}_{ci}$ to their correspondences on the CAD model, denoted by $\underline{\mathbf{p}}_{mi}$. First, the CAD model must be updated with the most recent state estimate $\boldsymbol{\rho}_k$. The estimated correspondences can then be computed as the closest point on the CAD model. The CAD model is broken into basic 2D primitives such as lines, arcs, and points, analogous to the DXF format commonly used in industry. For each entity, geometric formulae exist to compute these closest points analytically; however, in the current work, these are omitted for brevity.

D. Iterative Workpiece Registration Refinement

We define the residuals of $\underline{\tilde{\mathbf{p}}}_i$ and $\underline{\mathbf{p}}_{mi}$ as $\underline{\mathbf{e}}_i = \underline{\tilde{\mathbf{p}}}_{ci} - \underline{\mathbf{p}}_{mi}$. Iterative refinement is accomplished by minimizing the non-linear quadratic cost function

$$L(\boldsymbol{\rho}) = \tfrac{1}{2}\boldsymbol{\rho}^\mathsf{T}\mathbf{P}_0^{-1}\boldsymbol{\rho} + \tfrac{1}{2}\sum_{i=1}^{n} \underline{\mathbf{e}}_i^\mathsf{T}\, \mathbf{S}_i^{-1}\, \underline{\mathbf{e}}_i, \tag{7}$$

where $\mathbf{P}_0 \in \mathbb{R}^{4\times4}$ and $\mathbf{S}_i \in \mathbb{R}^{2\times2}$ are positive definite matrices of the initial state covariance and the measurement covariance of the i^{th} sample $\mathbf{e}_i$, respectively. The first quadratic term acts as an L_2 regularization term, drawing the solution towards the initial state estimate in the absence of better data, and the second term minimizes the squared norm of the residual vectors. Each term is weighted by its respective reciprocal covariance ($\mathbf{P}_0^{-1}$ or $\mathbf{S}_i^{-1}$), ensuring that the resulting minimum is optimal, in the sense of minimizing the variance across all information sources [36].

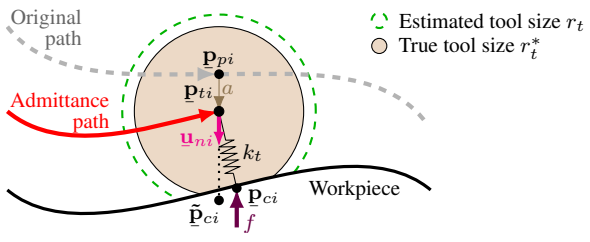


Fig. 5. Visualization of the SRAM tool-workpiece contact model.

Iterative refinement of $L(\boldsymbol{\rho})$ can be accomplished using a second-order Newton step update at each controller cycle to reach the stationary optimal point where $\nabla L(\boldsymbol{\rho}) = \mathbf{0}$, as

$$\boldsymbol{\rho}_k = \boldsymbol{\rho}_{k-1} - \left[\nabla^2 L(\boldsymbol{\rho})\right]^{-1} \nabla L(\boldsymbol{\rho}), \tag{8}$$

where the associated gradient and Hessian are

$$\begin{aligned}\nabla L(\boldsymbol{\rho}) &= \mathbf{P}_0^{-1}\boldsymbol{\rho} + \sum_{i=1}^{n} \nabla \underline{\mathbf{e}}_i^\mathsf{T}\, \mathbf{S}_i^{-1}\, \underline{\mathbf{e}}_i, \\ \nabla^2 L(\boldsymbol{\rho}) &= \mathbf{P}_0^{-1} + \sum_{i=1}^{n} \left[\nabla \underline{\mathbf{e}}_i^\mathsf{T} \mathbf{S}_i^{-1} \nabla \underline{\mathbf{e}}_i + \nabla^2 \underline{\mathbf{e}}_i\, \mathbf{S}_i^{-1} \underline{\mathbf{e}}_i\right].\end{aligned} \tag{9}$$

Eq. (9) requires that the residual Jacobian $\nabla \mathbf{e}_i$ and Hessian $\nabla^2 \mathbf{e}_i$ be computed. These equations are omitted for brevity; they can be derived analytically or estimated using a finite-differences approach.

E. Path Correction

As the state $\boldsymbol{\rho}_k$ is updated based on new observations from the real-time process, the original path can be corrected online with a translational adjustment $\mathbf{d}_{\text{cor}}$ and a rotational adjustment $\mathbf{R}_{\text{cor}}$, visualized previously in Fig. 4a, as

$$\mathbf{p}_{p,\text{cor}} = \mathbf{p}_p + \mathbf{d}_{\text{cor}}, \qquad \mathbf{u}_{n,\text{cor}} = \mathbf{R}_{\text{cor}}\mathbf{u}_n, \tag{10}$$

where the 2D vector $\underline{\mathbf{d}}_{\text{cor}}$ describes a translation $\delta\underline{\mathbf{d}} = \begin{bmatrix}\delta x & \delta y\end{bmatrix}^\mathsf{T}$, a rotation by θ, and a translation of δr along the admittance axis, as

$$\underline{\mathbf{d}}_{\text{cor}} = \delta\underline{\mathbf{d}} + \left(\underline{\mathbf{R}}_\theta\, \underline{\mathbf{p}}_p - \underline{\mathbf{p}}_p\right) - \delta r\, \underline{\mathbf{R}}_\theta\, \underline{\mathbf{u}}_n, \tag{11}$$

and the 3D vector $\mathbf{d}_{\text{cor}}$ results by projecting $\underline{\mathbf{d}}_{\text{cor}}$ into 3D space as in (4). The rotation matrix $\mathbf{R}_{\text{cor}}$ is similarly defined as the 3D projection of the 2D rotation matrix $\mathbf{R}_\theta$.

In applying the path corrections, it is necessary to subtract a velocity correction term $\dot{a}_{\text{cor}}$ to the admittance controller. Otherwise, the instantaneous path adjustments will be "felt" by the admittance controller and result in undesirable forces applied to the part. By chain rule, this velocity adjustment is

$$\dot{a}_{\text{cor}} = \frac{\partial a}{\partial t} = \left(\frac{\partial a}{\partial \boldsymbol{\rho}}\right)^\mathsf{T} \left(\frac{\partial \boldsymbol{\rho}}{\partial t}\right) = \nabla a^\mathsf{T}\, \dot{\boldsymbol{\rho}}, \tag{12}$$

where for a sample time t_s, the speed $\dot{\boldsymbol{\rho}} = (\boldsymbol{\rho}_k - \boldsymbol{\rho}_{k-1})/t_s$, and ∇a is computed by differentiating the projection of $\mathbf{d}_{\text{cor}}$ onto the corrected admittance axis $\underline{\mathbf{R}}_\theta\, \underline{\mathbf{u}}_n$ as

$$\begin{aligned}\nabla a &= \nabla\left((\underline{\mathbf{R}}_\theta\, \underline{\mathbf{u}}_n)^\mathsf{T}\, \mathbf{d}_{\text{cor}}\right) \\ &= \left[\left(\underline{\mathbf{R}}_\theta\, \underline{\mathbf{u}}_n\right)^\mathsf{T} \quad \left(\underline{\mathbf{R}}'_\theta\, \underline{\mathbf{u}}_n\right)^\mathsf{T}\left(\delta\underline{\mathbf{d}} - \underline{\mathbf{p}}_p\right) \quad -1\right]^\mathsf{T},\end{aligned} \tag{13}$$

where $\mathbf{R}'_\theta = \partial\mathbf{R}_\theta/\partial\theta$ is the analytic derivative of $\mathbf{R}_\theta$.

F. Covariance Estimation

A key insight of the SRAM framework is that the covariance of the workpiece registration and tool, $\tilde{\mathbf{P}}$, can be used to adaptively tune the admittance controller. This covariance matrix quantifies the uncertainty in the current state, *given*

the algorithm's "observations", e.g., either the initial state or the residual samples. Let $\mathbf{z}$ denote this set of generalized "observations." Then, the state covariance could be computed by linearly projecting the observation covariance as

$$\tilde{\mathbf{P}} = \mathrm{cov}(\boldsymbol{\rho}) = \left(\partial\boldsymbol{\rho}/\partial\mathbf{z}\right)\,\mathrm{cov}(\mathbf{z})\,\left(\partial\boldsymbol{\rho}/\partial\mathbf{z}\right)^{\mathsf{T}}. \tag{14}$$

While the observation covariances $\mathbf{P}_0$ and $\mathbf{S}_i$ are known, this formula presents problems as the partial derivative term $\partial\boldsymbol{\rho}/\partial\mathbf{z}$ cannot be readily computed, and there exists only an implicit relationship between $\mathbf{z}$ and $\boldsymbol{\rho}$ defined by the optimization of $L(\boldsymbol{\rho})$. However we can invoke the Implicit Function Theorem to derive the required term.

Theorem 1 (Implicit Function Theorem): Reworded from [37], [38]. Let $F : \mathbb{R}^h \times \mathbb{R}^n \to \mathbb{R}^h$ be a C^1 function with coordinates $(\mathbf{x}, \mathbf{y})$, such that at a point $(\mathbf{x}_0, \mathbf{y}_0)$

$$F(\mathbf{x}, \mathbf{y}) = \mathbf{0}, \qquad \tfrac{\partial F}{\partial \mathbf{y}}(\mathbf{x}_0, \mathbf{y}_0) \text{ is invertible.} \tag{15}$$

Then, there exists a neighborhood $U \subset \mathbb{R}^h$ of $\mathbf{x}_0$ such that there exists a unique C^1 mapping $g : U \to \mathbb{R}^n$ with $g(\mathbf{x}_0) = \mathbf{y}_0$, $F\big(\mathbf{x}, g(\mathbf{x})\big) = \mathbf{0}$ and partial derivatives

$$\tfrac{\partial g}{\partial \mathbf{y}}(\mathbf{x}) = -\big[\tfrac{\partial F}{\partial \mathbf{y}}\big(\mathbf{x}, g(\mathbf{x})\big)\big]^{-1}\,\big[\tfrac{\partial F}{\partial \mathbf{x}}\big(\mathbf{x}, g(\mathbf{x})\big)\big]. \tag{16}$$

Conveniently, we note that the relationship between $\mathbf{z}$ and $\boldsymbol{\rho}$ is defined by the first-order optimality condition $\nabla L(\boldsymbol{\rho}) = \mathbf{0}$, and thus fulfills (15), provided $\nabla^2 L(\boldsymbol{\rho})$ is invertible. Thus, the partial derivative term $\partial\boldsymbol{\rho}/\partial\mathbf{z}$ is given by (16). Substitution of the appropriate terms for both observations into (16), then again into (14) and simplifying yields the surprisingly elegant closed-form equation

$$\tilde{\mathbf{P}} = \big[\nabla^2 L(\boldsymbol{\rho})\big]^{-1}\big[\mathbf{P}_0^{-1} + \sum\nolimits_i \nabla\mathbf{e}_i{}^{\mathsf{T}}\,\mathbf{S}_i^{-1}\,\nabla\mathbf{e}_i\big]\big[\nabla^2 L(\boldsymbol{\rho})\big]^{-1}, \tag{17}$$

where $\nabla^2 L(\boldsymbol{\rho})$ and $\nabla\mathbf{e}_i$ are as in Section III-D, and all quantities are computed using the current state estimate, $\boldsymbol{\rho}_k$.

This approach of covariance estimation has been used in similar applications such as vision systems or ICP [39], [40] with success and is known to be reliable in providing covariance for algorithmic estimations. Interestingly, this estimate of $\tilde{\mathbf{P}}$ makes no assumptions about the implementation of the SRAM algorithm, nor the linearity of the problem. The sole assumption is that the covariances $\mathbf{P}_0$ and $\mathbf{S}_i$ are uncorrelated, allowing their reciprocals to be summed in (17). However, if the complete observation covariance were known, it could be readily substituted to yield an exact result.

G. Admittance Adaptation

The optimal behavior of the admittance algorithm is to respond quickly when the part location is unknown, but to slow and become more stable as the certainty of the registration improves. The online state covariance estimate $\tilde{\mathbf{P}}$ gives an ideal tool for this modulation. By projecting the state covariance onto the current admittance axis as

$$\sigma_a^2 = \mathrm{var}(a) = \nabla a^{\mathsf{T}}\,\tilde{\mathbf{P}}\,\nabla a, \tag{18}$$

with ∇a as in (13), we obtain σ_a^2—the variance of the admittance axis itself. This projected quantity is ideal since it allows the admittance to be modulated based on the *local* admittance conditions rather than the global convergence of the state parameters. In other words, the admittance can be tuned based on knowledge of whether the admittance correction is known, rather than whether the whole part has been localized. This distinction allows for ideal admittance behavior, as demonstrated later in Section IV. The admittance controller is adaptively tuned with a scaling parameter γ_a, in the range 0–1, used in the modified controller

$$\ddot{a} = \tfrac{1}{m_a}\Big[\gamma_a(f_d - f) - b_a\dot{a} - k_a a\Big],\;\; a = \int\Big(\int \ddot{a}\,dt - \dot{a}_{\mathrm{cor}}\Big)dt. \tag{19}$$

The scaling parameter γ_a should be computed from σ_a^2, according to the application. In the current work, γ_a is computed according to the heuristic

$$\gamma_a = 1 - \big(1 - (\sigma_{a0}/\sigma_a)^{j_a}\big)\Big/\big(1 - \big(\sigma_{a0}/\sigma_{a,\min}\big)^{j_a}\big), \tag{20}$$

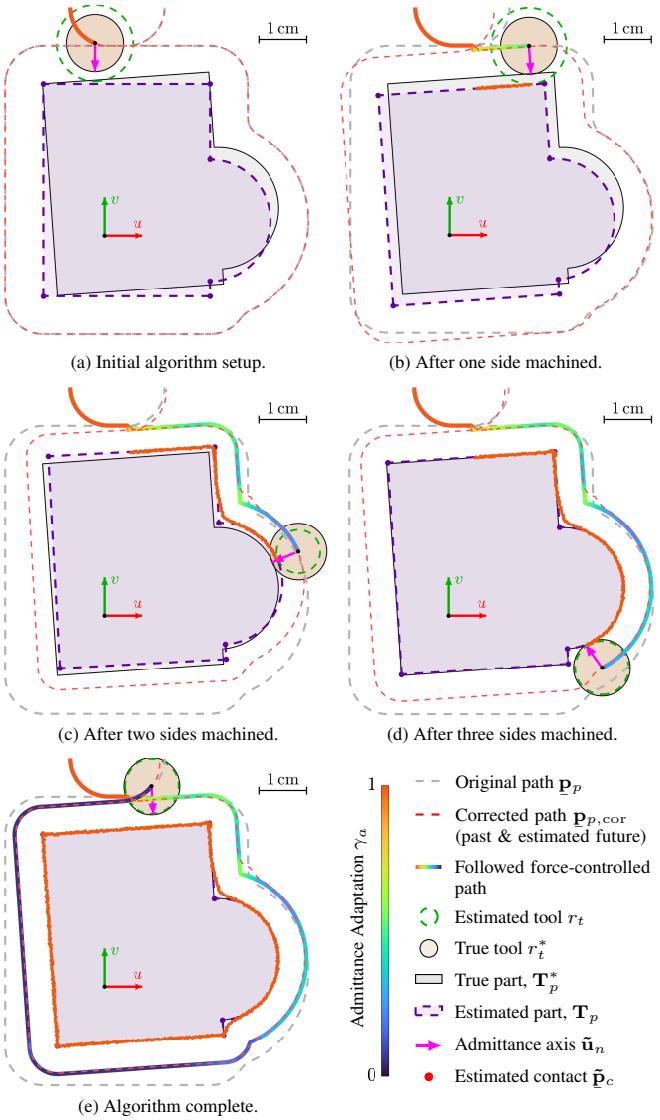
where σ_{a0} is the initial admittance standard error, $\sigma_{a,\min}$ is the minimum standard error (e.g., the error associated with no force control, and set according to the repeatability of the robot), and j_a is a tunable shape parameter. The result is then saturated within the range 0–1. While complex, this heuristic ensures that as σ_a approaches $\sigma_{a,\min}$, γ_a tends from one to zero, and the admittance becomes increasingly damped.

IV. Framework Validation

To validate the proposed framework, we simulate the algorithm in the contouring of a sample part. This simulation assumes that the robot follows the commanded path perfectly and that observed forces can be modeled as per (6), with parameter uncertainty and overlaid noise. These assumptions are not expected to result in significant loss in fidelity, as the SRAM adaption runs at rates slower than the robot bandwidth, and the force modeling and noise are informed from real-world testing. Therefore, this simulation can provide important insight and validation of the framework.

The contouring path is planned offline and fed into the simulation. The desired force f_d is $3\,\mathrm{N}$. Contact forces are computed with the tool stiffness k_t by geometrically calculating the tool interaction depth. Measured force noise is simulated after real-world testing as overlaid sinusoidal noise at the spindle frequency of $5000\,\mathrm{rpm}$ with an amplitude of $5\,\mathrm{N}$. The admittance controller is run at $1\,\mathrm{kHz}$, while the SRAM updates are performed slower at $25\,\mathrm{Hz}$. Admittance parameters are set to typical values of $m_a = 2\,\mathrm{kg}$, $b_a = 300\,\mathrm{Ns/m}$, and $k_a = 0\,\mathrm{N/m}$, and adaptive parameters of $\sigma_{a0} = 3\,\mathrm{mm}$, $\sigma_{a,\min} = 0.01\,\mathrm{mm}$, and $j_a = 0.5$. Tool diameter is $7/16$" ($11.1\,\mathrm{mm}$), and tool stiffness is simulated as $k_t = 20\,\mathrm{kN/m}$ with 20% assumed parameter error. The simulated workpiece is shown in Fig. 6a, and the initial estimate is perturbed by $\boldsymbol{\rho} = \begin{bmatrix}2 & 1 & 4 & -2\end{bmatrix}$ with covariance $\mathbf{P}_0 = \mathrm{diag}\begin{bmatrix}3^2 & 3^2 & 3^2 & 2^2\end{bmatrix}$ (in mm and deg), and the measurement covariance is set to $\mathbf{S}_i = \mathrm{diag}\begin{bmatrix}0.1^2 & 0.1^2\end{bmatrix}\,\mathrm{mm}^2$.

Fig. 6b-e shows the workpiece registration estimate, tool estimate, and original as well as updated paths as the tool moves around the workpiece in this simulation. Fig. 7 (top) shows the live admittance covariance estimate and associated tuning parameter γ_a, and Fig. 7 (middle) plots the



(a) Initial algorithm setup. (b) After one side machined. (c) After two sides machined. (d) After three sides machined. (e) Algorithm complete.

Fig. 6. Snapshots of the convergence of the SRAM algorithm during the edge polishing of a sample workpiece.

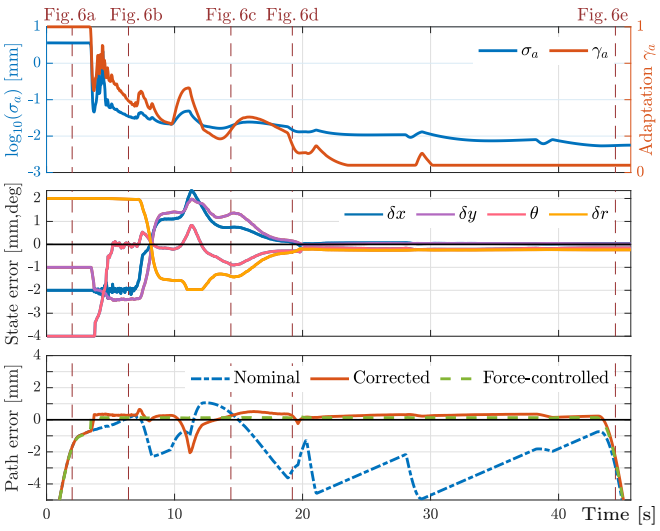


Fig. 7. *Top*: Admittance variance estimate σ_a and associated adaptation γ_a. *Middle*: Convergence of the state parameters $\boldsymbol{\rho}$. *Bottom*: Projected path error of the commanded path with and without the SRAM corrections.

online state estimate error. Finally, Fig. 7 (bottom) shows the projected path error, e.g., the correction that must be supplied by the admittance controller, and the magnitude of which determines the quality of the part and speed of the process. Initially, the algorithm has a poor estimate of the workpiece location and tool size. After one side of the workpiece is machined, the state estimate improves—the angle of the part is known, and while neither the horizontal workpiece position nor the tool size are known, their *sum*, e.g., the commanded tool center, is correct. This fact is reflected in the covariance estimate σ_a, which begins high, but immediately lowers as knowledge of the horizontal edge is obtained, despite the fact that none of the state variables are known. Thus, we observe the advantage of using the local admittance covariance in the modulation of γ_a—the algorithm can identify that the tool center is correct and that the admittance can be stiffened, even though it does not know δy or δr individually. Similarly, after the tool passes the first corner in Fig. 6c, the algorithm still has not identified δx, δy, or δr, since the workpiece could be shifted diagonally with a different tool size, without the observations changing. However, the admittance is dampened since the tool position is known to be correct. Only after the tool wraps around the part, in Fig. 6d, does the algorithm have sufficient data to converge fully to the correct state variables.

In the entire process, the admittance controller is required three times—at the start, to identify the vertical offset, after the first corner, to identify the horizontal offset, and at the start and around the right lobe to learn the tool radius. In both cases, the need for rapid admittance corrections is properly identified by the SRAM algorithm, and the controller tuning is adjusted accordingly. Overall, Fig. 6 and Fig. 7 show that the corrections required by the controller are significantly reduced compared to the nominal case, and all the major errors in the corrected path are caused by poor knowledge of the initial state while the algorithm is converging. Thus, we can note that this improvement would be even more significant on more complex workpieces.

V. Conclusion

The novel SRAM framework is introduced to leverage real-time acquired data to improve workpiece registration online in contouring applications, reducing path errors. Simultaneously, a live covariance estimate is leveraged to adaptively tune the controller to behave optimally based on local process conditions. The method is demonstrated in simulation on a sample part and is shown to significantly reduce path errors, while correctly predicting optimal modulations to the admittance controller. Thus, the controller is required to compensate for less path error, and is dynamically tuned to quickly correct offsets when needed, and behave rigidly otherwise—allowing for more effective removal of imperfections, smoother surface finish, and improved peripheral accuracy. SRAM is proposed as a generalized framework for improving the performance of contouring tasks, and could be readily extended to other applications. Future work will test this algorithm on hardware in real-world robotic contouring.

REFERENCES

[1] International Federation of Robotics, "World Robotics 2019, Statistical Yearbook," Tech. Rep., 2019.

[2] A. Karim and A. Verl, "Challenges and obstacles in robot-machining," *2013 44th Int. Symp. Robot. ISR 2013*, 2013.

[3] I. F. Onstein, O. Semeniuta, and M. Bjerkeng, "Deburring Using Robot Manipulators: A Review," *2020 3rd Int. Symp. Small-Scale Intell. Manuf. Syst. SIMS 2020*, 2020.

[4] A. Verl, A. Valente, S. Melkote, C. Brecher, E. Ozturk, and L. T. Tunc, "Robots in machining," *CIRP Ann.*, vol. 68, no. 2, pp. 799–822, 2019.

[5] M. Jonsson, A. Stolt, A. Robertsson, S. von Gegerfelt, and K. Nilsson, "On force control for assembly and deburring of castings," *Prod. Eng.*, vol. 7, no. 4, pp. 351–360, 2013.

[6] S. A. Niknam, B. Davoodi, J. P. Davim, and V. Songmene, "Mechanical deburring and edge-finishing processes for aluminum parts—a review," *Int. J. Adv. Manuf. Technol.*, vol. 95, no. 1-4, pp. 1101–1125, mar 2018.

[7] H. Zhang, H. Chen, N. Xi, G. Zhang, and J. He, "On-line path generation for robotic deburring of cast aluminum wheels," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 2400–2405, 2006.

[8] L. Ma, P. Bazzoli, P. M. Sammons, R. G. Landers, and D. A. Bristow, "Modeling and calibration of high-order joint-dependent kinematic errors for industrial robots," *Robot. Comput. Integr. Manuf.*, vol. 50, no. September 2017, pp. 153–167, 2018.

[9] G. Luo, L. Zou, Z. Wang, C. Lv, J. Ou, and Y. Huang, "A novel kinematic parameters calibration method for industrial robot based on Levenberg-Marquardt and Differential Evolution hybrid algorithm," *Robot. Comput. Integr. Manuf.*, vol. 71, no. September 2020, p. 102165, 2021.

[10] J. Hu and P. R. Pagilla, "Dual-edge robotic gear chamfering with registration error compensation," *Robot. Comput. Integr. Manuf.*, vol. 69, no. December 2020, p. 102082, 2021.

[11] E. Villagrossi, C. Cenati, N. Pedrocchi, M. Beschi, and L. Molinari Tosatti, "Flexible robot-based cast iron deburring cell for small batch production using single-point laser sensor," *Int. J. Adv. Manuf. Technol.*, vol. 92, no. 1-4, pp. 1425–1438, sep 2017.

[12] W. L. Li, H. Xie, G. Zhang, S. J. Yan, and Z. P. Yin, "3-D Shape Matching of a Blade Surface in Robotic Grinding Applications," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 5, pp. 2294–2306, 2016.

[13] J. R. Diaz Posada, U. Schneider, S. Pidan, M. Geravand, P. Stelzer, and A. Verl, "High accurate robotic drilling with external sensor and compliance model-based compensation," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016-June, pp. 3901–3907, 2016.

[14] J. Posada, R. Diaz, S. Kumar, A. Kuss, U. Schneider, M. Drust, T. Dietz, and A. Verl, "Automatic programming and control for robotic deburring," in *Proc. ISR 2016 47st Int. Symp. Robot.* VDE, 2016, pp. 1–8.

[15] H. Kosler, U. Pavlovčič, M. Jezeršek, and J. Možina, "Adaptive Robotic Deburring of Die-Cast Parts with Position and Orientation Measurements Using a 3D Laser-Triangulation Sensor," *Stroj. Vestnik/Journal Mech. Eng.*, vol. 62, no. 4, pp. 207–212, 2016.

[16] G. Xiong, Z. L. Li, Y. Ding, and L. M. Zhu, "A closed-loop error compensation method for robotic flank milling," *Robot. Comput. Integr. Manuf.*, vol. 63, no. November 2019, p. 101928, 2020.

[17] M. Dehghani, R. A. McKenzie, R. A. Irani, and M. Ahmadi, "Robot-mounted sensing and local calibration for high-accuracy manufacturing," *Robot. Comput. Integr. Manuf.*, vol. 79, no. April 2022, p. 102429, 2022.

[18] K. M. Murphy, R. J. Norcross, and F. M. Proctor, "CAD directed robotic deburring," *Proc. Second Int. Symp. Robot. Manuf. Res. Educ. Appl. Albuquerque, NM*, 1988.

[19] N. Asakawa, K. Toda, and Y. Takeuchi, "Automation of chamfering by an industrial robot; for the case of hole on free-curved surface," *Robot. Comput. Integr. Manuf.*, vol. 18, no. 5-6, pp. 379–385, 2002.

[20] H. C. Song and J. B. Song, "Precision robotic deburring based on force control for arbitrarily shaped workpiece using CAD model matching," *Int. J. Precis. Eng. Manuf.*, vol. 14, no. 1, pp. 85–91, 2013.

[21] M. Jinno, M. Uenohara, J. Oaki, and K. Tatsuno, "Teaching-less robot system for finishing workpieces of various shapes using force control and computer vision," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 1, pp. 573–578, 1999.

[22] A. Kuss, M. Drust, and A. Verl, "Detection of Workpiece Shape Deviations for Tool Path Adaptation in Robotic Deburring Systems," *Procedia CIRP*, vol. 57, pp. 545–550, 2016.

[23] B. Mei, Z. Liang, W. Zhu, and Y. Ke, "Positioning variation synthesis for an automated drilling system in wing assembly," *Robot. Comput. Integr. Manuf.*, vol. 67, no. July 2020, p. 102044, 2021.

[24] Z. Lai, R. Xiong, H. Wu, and Y. Guan, "Integration of Visual Information and Robot Offline Programming System for Improving Automatic Deburring Process," *2018 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2018*, pp. 1132–1137, 2018.

[25] F. Leo Princely and T. Selvaraj, "Vision assisted robotic deburring of edge burrs in cast parts," *Procedia Eng.*, vol. 97, pp. 1906–1914, 2014.

[26] T. F. Lu and G. C. Lin, "An on-line relative position and orientation error calibration methodology for workcell robot operations," *Robot. Comput. Integr. Manuf.*, vol. 13, no. 2, pp. 89–99, 1997.

[27] M. Dinham and G. Fang, "Autonomous weld seam identification and localisation using eye-in-hand stereo vision for robotic arc welding," *Robot. Comput. Integr. Manuf.*, vol. 29, no. 5, pp. 288–301, 2013.

[28] M. Amersdorfer, J. Kappey, and T. Meurer, "Real-time freeform surface and path tracking for force controlled robotic tooling applications," *Robot. Comput. Integr. Manuf.*, vol. 65, no. August 2019, p. 101955, 2020.

[29] J. E. Solanes, L. Gracia, P. Muñoz-Benavent, A. Esparza, J. Valls Miro, and J. Tornero, "Adaptive robust control and admittance control for contact-driven robotic surface conditioning," *Robot. Comput. Integr. Manuf.*, vol. 54, pp. 115–132, dec 2018.

[30] D. E. Whitney, "Force feedback control of manipulator fine motions," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 99, no. 2, pp. 91–97, 1977.

[31] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified impedance and admittance control," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 554–561, 2010.

[32] B. Li, "A review of tool wear estimation using theoretical analysis and numerical simulation technologies," *Int. J. Refract. Met. Hard Mater.*, vol. 35, pp. 143–151, 2012.

[33] K. Schützer, E. Abele, and S. Güth, "Simulation-based deburring tool and process development," *CIRP Ann. - Manuf. Technol.*, vol. 64, no. 1, pp. 357–360, 2015.

[34] D. Systemes, "Solidworks CAM," 2022. [Online]. Available: https://www.solidworks.com/product/solidworks-cam

[35] Siemens, "NX for Manufacturing," 2022. [Online]. Available: https://www.plm.automation.siemens.com/global/en/products/nx/nx-for-manufacturing.html

[36] T. P. Ryan, *Modern Regression Methods*, 2nd ed. John Wiley and Sons, 2009.

[37] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. McGraw Hill, 1976.

[38] O. de Oliveira, "The implicit and the inverse function theorems: Easy proofs," *Real Anal. Exch.*, vol. 39, no. 1, pp. 207–218, 2013.

[39] A. K. R. Chowdhury and R. Chellappa, "Stochastic Approximation and Rate-Distortion Analysis for Robust Structure and Motion Estimation," *Int. J. Comput. Vis.*, vol. 55, no. 1, pp. 27–53, 2003.

[40] A. Censi, "An accurate closed-form estimate of ICP'S covariance," *Proc. - IEEE Int. Conf. Robot. Autom.*, no. April, pp. 3167–3172, 2007.