Human-like Handwriting Using a 7-DOF Robot Manipulator

Arnav Srikanth
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA
asrikan2@andrew.cmu.edu

Gabriel Rodriguez
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA
gar2@andrew.cmu.edu

***Abstract*—Autopen systems, specialized machines to reproduce handwritten signatures in large volumes, have existed for decades. By constraining the end-effector to three degrees-of-freedom (DOF), the specialized hardware simplifies the handwriting replication problem. This paper explores the use of a 7 DOF robotic manipulator to perform the same tasks while not requiring the specialized hardware used by autopen systems. We propose the use of a computer vision-based trajectory generation approach that models not only the pen stroke trajectory but also the desired stroke thickness. To track the generated trajectories, a PD force control is used to modulate the pen stroke trajectory for thickness tracking, and the modulated trajectory is tracked by a model predictive controller using linearized task-space kinematics. The resulting pipeline achieves a success rate of 82.35% on handwriting excerpts from our custom dataset, which includes both print and cursive text with varying stroke thickness.**

***Index Terms*—manipulation, tool use, computer vision, optimal control**

I. Introduction

The replication of human handwriting by machines is a capability with a rich history, dating back to early pantographs and evolving into modern "autopen" systems. These automated signature machines are widely utilized for high-volume signing tasks in government, celebrity correspondence, and executive administration [1]. Traditionally, these systems rely on specialized hardware configurations constrained to three degrees-of-freedom (DOF), operating primarily as varying planar plotters with a binary pen-up/pen-down mechanism [2]. Although effective for their specific intended use, the reliance on dedicated, low-DOF hardware limits the flexibility and application scope of automated handwriting replication.

In recent years, the ubiquity of multi-degree-of-freedom robotic manipulators in research and industrial settings has presented an opportunity to generalize handwriting tasks beyond specialized hardware. However, replicating the nuance of human handwriting, specifically the variation in stroke width and pressure, remains a complex challenge for general-purpose robots. Unlike the constrained kinematics of an autopen, a 7-DOF robotic manipulator introduces increased complexity that requires sophisticated control strategies to ensure legible and faithful reproduction.

Source code containing all data, experiments, and analysis is made available at: `https://github.com/gabearod2/calligraphy`

This paper proposes a framework for automated handwriting using a Franka Emika Panda 7-DOF robotic manipulator [3]. We address the replication problem through a holistic approach that integrates computer vision with control theory.
The objectives of this project are summarized as follows:

1) **Computer Vision-Based Trajectory Generation:** Develop a system that extracts not only the planar geometry of a signature but also encodes desired stroke thickness into the trajectory.
2) **Hybrid Control Architecture:** Implement a Proportional-Derivative (PD) force controller to modulate the vertical (z-axis) trajectory for thickness tracking.
3) **Model Predictive Control (MPC) Tracking:** Utilize an MPC framework using linearized task-space kinematics to track the modulated trajectory with high precision.

II. Methods

A. Trajectory Generation

To enable precise force modulation, a computer vision pipeline was developed to simultaneously extract the spatial trajectory $\boldsymbol{w}$ and the local stroke thickness $\boldsymbol{\delta}$ from the input image. The system architecture is illustrated in Fig. 1.

The processing sequence begins with image acquisition and binary thresholding to maximize the contrast between the handwriting stroke and the background. A Euclidean Distance Transform (EDT) is then applied to the binarized image. Let $\Omega \subset \mathbb{Z}^2$ denote the discrete image domain. We define $\mathcal{S} \subset \Omega$ as the set of foreground pixels representing the handwriting stroke, and $\mathcal{B} = \Omega \setminus \mathcal{S}$ as the set of background pixels. The Euclidean Distance Transform, $D(u)$, for any pixel $v \in \mathcal{S}$ is defined as the minimum Euclidean distance to the complement set $\mathcal{B}$:

$$D(u) = \min_{v \in \mathcal{B}} \|u - v\|_2 \tag{1}$$

where $\|\cdot\|_2$ denotes the L_2 norm. Consequently, the thickness parameter δ at a specific trajectory point is proportional to $2 \cdot D(u)$. This operation calculates the distance from every pixel within the stroke to the nearest zero-valued (background) pixel, effectively creating a map where pixel intensity corresponds to local stroke width [4].

To derive the path for the pen tip, the binarized image is processed using the Zhang-Suen thinning algorithm [5]. This morphological operation iteratively erodes the stroke boundaries to produce a skeletonized, single-pixel-wide representation of the trajectory. The resulting skeleton is segmented into discrete contours, which are sorted spatially to enforce a left-to-right writing sequence [6]. Finally, the system traverses the ordered contours, extracting the 2D Cartesian coordinates to form the trajectory $\boldsymbol{w}$. At each coordinate, the corresponding thickness value is sampled from the EDT map to generate the thickness profile $\boldsymbol{\delta}$. These parameters are subsequently output to the control architecture.

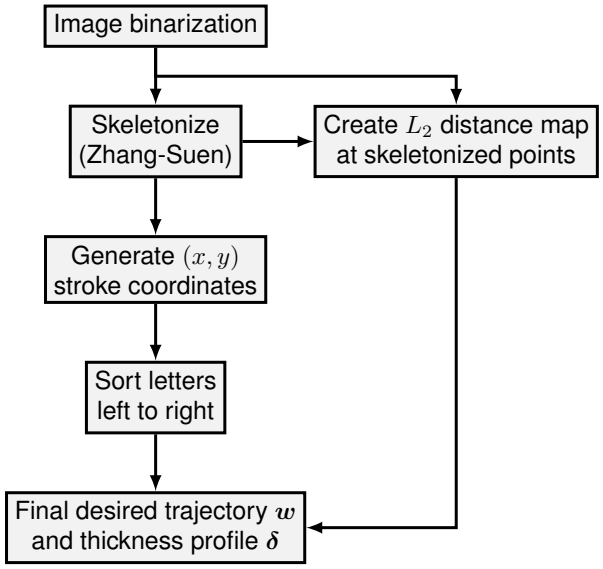

Fig. 1. Computer vision pipeline used for trajectory generation.

B. Trajectory Following

We aim to compute joint position commands that allow the Franka arm to follow both the desired writing trajectory $\boldsymbol{w}$ and the associated thickness profile $\boldsymbol{\delta}$. Assuming that thickness corresponds to how firmly the pen-tip is pressed into the writing pad [7], we propose a controller with two components:

1) **Outer loop force controller:** Maps the desired thickness profile $\boldsymbol{\delta}$ to a modified trajectory $\hat{\boldsymbol{w}}$ using force control.
2) **Inner loop kinematic controller:** Maps the modified trajectory to positional joint commands $\mathbf{q}_{\text{cmd}}$ by optimizing over a linearized task-space dynamics model.

1) Outer-Loop Force Control: In writing, we assume that thicker strokes are produced when the writing utensil applies a greater normal force. To represent this behavior, we model thickness using a spring model. We first map a desired thickness to a desired force using

$$f_\delta = k_\delta \delta, \tag{2}$$

and we model the estimated contact force between the pen and the pad using

$$f_z = k_z(z_{\text{pen-tip}} - z_{\text{pad}}). \tag{3}$$

In simulation, we measure the actual normal force f_n between the pen and the writing pad when it is available and use f_z when it is not. This allows a smooth transition between contact and non-contact modes. Additionally, this implementation uses ground-truth contact forces between the bodies as opposed to force-torque sensors in the simulator.

Thus, for a prediction horizon, we can define the error vector

$$\mathbf{e}_\delta = \begin{bmatrix} 0 & 0 & k_\delta \boldsymbol{\delta} - f_n \end{bmatrix}^\top \tag{4}$$

where $\boldsymbol{\delta}$ is the desired thickness profile over the current prediction horizon. Finally, the trajectory modulation for the given horizon can be defined as the PD update rule:

$$\hat{\boldsymbol{w}} = \boldsymbol{w} - k_p \mathbf{e}_\delta - k_d \dot{\mathbf{e}}_\delta, \tag{5}$$

where k_p and k_d are tuning constants.

2) Inner-Loop Kinematic Control:

a) Linearized Task-Space Dynamics Model: Following [8], we discretize the kinematic joint model using

$$\dot{q}_i = \frac{q_i - q_{i-1}}{\Delta t}, \quad \ddot{q}_i = \frac{q_i - 2q_{i-1} + q_{i-2}}{\Delta t^2}. \tag{6}$$

For a finite-time horizon $\mathbf{T} = [t_i, t_f]$, we can vertically concatenate the joint positions, velocities, and accelerations, and rewrite the kinematic relationships as

$$\dot{\mathbf{q}} = \mathbf{S}_v \mathbf{q}, \quad \ddot{\mathbf{q}} = \mathbf{S}_a \dot{\mathbf{q}}, \tag{7}$$

where finite-difference matrices are

$$\mathbf{S}_v = \frac{1}{\Delta t} \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots \\ 0 & 1 & -1 & 0 & \cdots \\ \vdots & & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix}, \tag{8}$$

$$\mathbf{S}_a = \frac{1}{\Delta t^2} \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & -2\ 1 \end{bmatrix}. \tag{9}$$

We also define a "task function" $g : \mathbb{R}^n \mapsto \mathbb{R}^{n_g}$ where

$$w = g(q), \quad \Delta w \approx J \Delta q, \tag{10}$$

and $J \in \mathbb{R}^{n_g \times n}$ is the task-space Jacobian. In our case, g corresponds to the forward kinematics of the Franka robot in simulation. We assume g is C^1 for later use in optimization.

b) MPC Formulation: In each control cycle, we construct a nominal joint trajectory $\hat{\mathbf{q}} = \{\hat{q}_0, \ldots, \hat{q}_{n_p}\}$ for the corresponding task-space trajectory $\hat{\boldsymbol{w}} = \{\hat{w}_0, \ldots, \hat{w}_{n_p}\}$ by computing the inverse kinematics (IK) for each $\hat{w}_k$. We replace the first element of the prediction horizon with the current measured state q and the pen-tip position w to ensure that the MPC accounts for the current state. The simulator also allows us to gather the task-space (linear and angular) Jacobians $\hat{J}_k$ at each point in the nominal joint space trajectory $\hat{q}_k$.

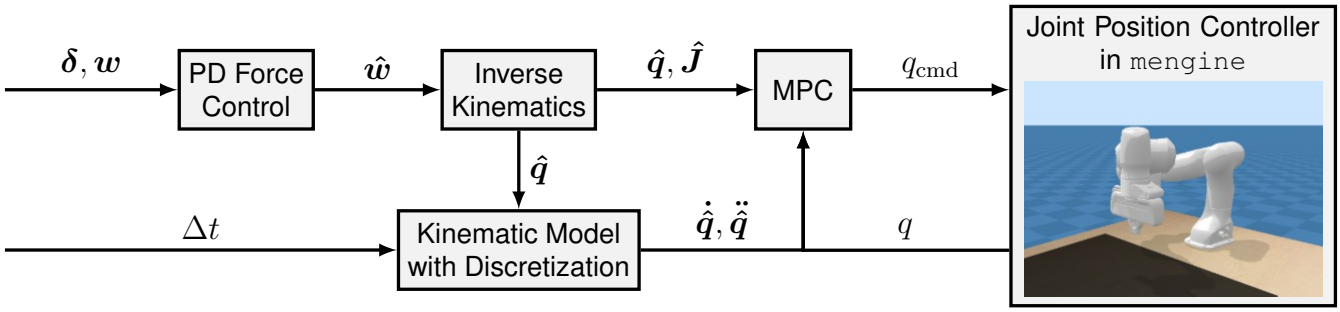

Fig. 2. The proposed control architecture for handwriting trajectory tracking combining thickness tracking force control, inverse kinematics for nominal trajectory generation, and a linearized MPC with joint-space position execution in `mengine`.

We then formulate the optimization problem for the kinematic MPC approach as follows:

$$\begin{aligned}\min_{\mathbf{q}} \mathcal{J} = &\sum_{k=i}^{i+n_p} \left(\hat{J}_k(\hat{q}_k - q_k)^\top\right) Q_e \left(\hat{J}_k(\hat{q}_k - q_k)\right) \\ &+ (\hat{q}_k - q_k)^\top Q_r (\hat{q}_k - q_k) \\ &+ \dot{q}_k^\top Q_v \dot{q}_k \\ &+ \ddot{q}_k^\top Q_a \ddot{q}_k, \\ \text{s.t.} \quad &\dot{q}_{\min} \le \dot{q}_k \le \dot{q}_{\max}, \\ &q_{\min} \le q_k \le q_{\max}, \qquad \forall k \in \{i, \dots, i+n_p\}.\end{aligned} \tag{11}$$

where $Q_e \in \mathbb{R}^{n_g \times n_g}$, $Q_r \in \mathbb{R}^{n \times n}$, $Q_v \in \mathbb{R}^{n \times n}$, and $Q_a \in \mathbb{R}^{n \times n}$ denote the weighting matrices for the tracking task error (position and orientation), joint-space regularization, velocity smoothing, and acceleration smoothing terms, respectively.

Linearization allows us to simplify this optimization program into the following quadratic program (QP):

$$\begin{aligned}\min_{\mathbf{q}} \mathcal{J} = &\mathbf{q}^\top \mathbf{Q} \mathbf{q} + 2\mathbf{p}^\top \mathbf{q} + \text{const}, \\ \text{s.t.} \quad &\dot{\mathbf{q}}_{\min} \le \mathbf{S}_v \mathbf{q} \le \dot{\mathbf{q}}_{\max}, \\ &\mathbf{q}_{\min} \le \mathbf{q} \le \mathbf{q}_{\max}.\end{aligned} \tag{12}$$

This simplification allows us to rapidly compute the optimal sequence of joint position commands over the prediction horizon. We take the first joint position command as q_{cmd} and pass it to the simulator's lower-level position controller.

III. Implementation Details

We implement the proposed framework within `mengine` [9]. A custom URDF model of a pen was developed for ease of grasping and to provide a rounded pen-tip. The pen is given a high friction coefficient for stable grasping, while the writing pad is modeled as a planar object with zero sliding friction to simulate a ballpoint pen-paper interaction.

The trajectory generation pipeline is packaged as a callable function that returns a set of 2D cartesian coordinates separated by character with accompanying thickness profile. Given the set of contours in the generated trajectory, the controller first uses IK to reach the first pen-tip contact. Upon contact, the MPC is executed with a prediction horizon $n_p = 10$, a control timestep of $\Delta t = 0.2$ [s], and padding the reference trajectory with its final point when the horizon contains fewer than n_p points.

To simulate writing, the implementation initializes a sphere of radius $r = f_n/k_\delta$ at the simulated contact point. Finally, to track the pen-tip pose and optimize over joint angles, the desired trajectory is also displaced by the resultant vector from the pen-tip to the end-effector by accessing the pen's orientation at each control cycle. This allows the controller to adjust to the grip of the pen.

IV. Experiments and Results

A. Trajectory Generation Evaluation

To validate the fidelity of the trajectory generation pipeline, a comparative analysis was conducted between the ground truth binarized image A and the reconstructed stroke geometry B on 17 different handwriting excerpts written on an iPad.

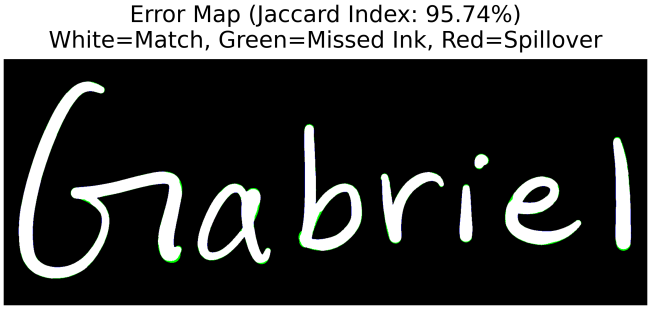

Fig. 3. Jaccard Index and Difference Map

Quantitatively, geometric congruence was measured using the Intersection over Union (IoU) metric, formally known as the Jaccard Index $J_I(A, B)$. This metric quantifies the overlap between the finite sample sets, defined as:

$$J_I(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad 0 \le J_I(A, B) \le 1, \tag{13}$$

where a value of 1 indicates perfect reconstruction and 0 indicates disjoint [10]. For the word *Gabriel*, the pipeline produced $J_I(A, B) = 0.9574$, which indicates excellent reconstruction. The quantitative results of the remainder of the dataset are shown in Table I.

TABLE I
PER-WORD TRAJECTORY GENERATION PERFORMANCE

Word	Jaccard Index
arnav_print	0.9641
arnav_cursive	0.9456
g_print	0.9680
gabriel_print	0.9562
gabriel_cursive	0.9174
manipulation_print	0.9534
manipulation_cursive	0.9481
mechanics_print	0.9523
mechanics_cursive	0.9463
motion_print	0.9527
motion_cursive	0.9338
symbol_print	0.9605
symbol_cursive	0.9221
talk_print	0.9637
talk_cursive	0.9620
virtual_print	0.9554
virtual_cursive	0.9628
Mean	**0.9508**

Qualitatively, a difference map was generated to visualize discrepancies. As illustrated in Fig 3, the reconstruction is overlaid onto the ground truth: white pixels denote perfect alignment (True Positives), green pixels indicate omitted stroke areas (False Negatives), and red pixels represent trajectory overshoot (False Positives).

B. Control Approach Ablation

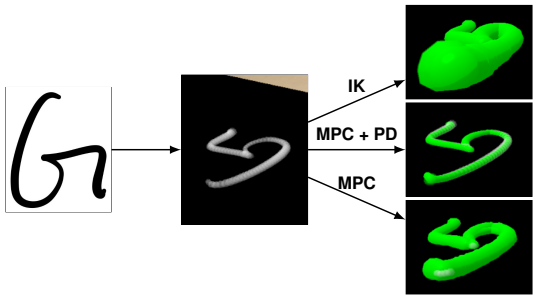

Fig. 4. Ablation comparison for the letter **G**.

To evaluate our control framework, we perform an ablation comparing: (i) a bare inverse kinematic approach, (ii) the linear MPC without force control, and (iii) the linear MPC with force control. All methods are given the same reference trajectory from the letter **G**. Fig 4 illustrates the qualitative results of the ablation study, while Table II reports the quantitative results. The proposed approach outperforms the more naive approaches by a wide margin. Although the other methods initially perform adequately, the thickness and position tracking diverge throughout the length of the trajectory.

C. Trajectory Following Evaluation

To evaluate the proposed trajectory-following architecture, we use our aforementioned custom dataset of 17 different handwritten excerpts. They were chosen to span a representative set of handwriting motions of varying curvature, stroke length, and thickness variation.

TABLE II
COMPARISON OF TRACKING PERFORMANCE ACROSS CONTROL METHODS FOR THE LETTER **G**

Method	RMSE_w	MaxErr_w	RMSE_δ	MaxErr_δ
IK	0.017204	0.037448	0.016401	0.052959
MPC	0.012239	0.019739	0.006387	0.014597
MPC + PD	**0.002234**	**0.004541**	**0.003132**	**0.005209**

All values are expressed in meters [m].

A varied dataset exposes the strengths and weaknesses of the controller. The primary failure mode observed in unsuccessful trials was degradation of the grasp on the pen until it fully slipped out the gripper's grasp. This was especially evident for words with thicker stroke profiles, such as `mechanics_print`. The higher thicknesses require higher normal forces, which generate greater reaction forces and torques at the grip of the pen.

Additionally, longer trajectories increased the rate of grasp failure. The excerpt `manipulation_cursive`, for example, saw general degradation of the grasp due to orientation changes in the pen grasp.

Together, these failure mode characteristics highlight the fragility of the pen-grasp. Despite this, as shown in Table III, we observe that for trajectories as long as 2.69 [m] and thicknesses up to 12.6 [mm], the tracking of the task-space remains excellent with an average $\text{RMSE}_w = 1.16$ [cm]. However, thickness tracking could be improved by further gain tuning or adding an integral gain with an average $\text{RMSE}_\delta = 3.5$ [mm].

TABLE III
PER-WORD TRACKING PERFORMANCE (MPC + PD CONTROLLER)

Word	RMSE_w [m]	RMSE_δ [m]	Success
arnav_print	0.006599	0.002727	Yes
arnav_cursive	0.006675	0.002201	Yes
g_print	0.002234	0.003132	Yes
gabriel_print	0.035215	0.006134	Yes
gabriel_cursive	0.008262	0.001614	Yes
manipulation_print	0.007883	0.006113	Yes
manipulation_cursive	—	—	No
mechanics_print	—	—	No
mechanics_cursive	0.049613	0.006741	Yes
motion_print	0.005574	0.002810	Yes
motion_cursive	0.005357	0.002691	Yes
symbol_print	—	—	No
symbol_cursive	0.008160	0.002659	Yes
talk_print	0.004921	0.004143	Yes
talk_cursive	0.006492	0.003211	Yes
virtual_print	0.006542	0.002905	Yes
virtual_cursive	0.013484	0.002217	Yes
Mean (successful only)	**0.0116**	**0.0035**	—
Success Rate	**82.35%**		

All values are expressed in meters [m].

V. CONCLUSION

In this report, we present a unified approach to writing using a 7-DOF robotic manipulator. By combining a computer vision pipeline with a hybrid control architecture, the system performs human-like handwriting for a variety of words and

writing styles in simulation. Experiments with 17 different words highlighted limitations due to grasp degradation, but showed good task-space tracking performance. These results highlight the difficulty of contact-rich force control even in simulated robotic environments, while validating the effectiveness of control theory in well-defined problems. Future improvements might include additional grasp stabilization, the use of force-torque sensors for normal force estimation, or the expansion of the force control architecture to account for grasp limitations.

REFERENCES

[1] C. Allison. (2025, Mar.) What is an 'autopen' and do Trump's claims about Joe Biden's use of it stack up? [Accessed: 2025-11-30]. [Online]. Available: https://www.abc.net.au/news/2025-03-18/what-is-an-autopen-and-trump-claims-examined/105064992

[2] N. S. Reddy, D. S. K. Kanth, K. Chandrakala, N. Keerthi, V. A. Reddy, and V. L. B. P. Reddy, "Design and development of automated writing machine," *International Journal of Recent Advances in Multidisciplinary Topics*, vol. 4, no. 2, pp. 66–70, February 2023. [Online]. Available: https://www.ijramt.com

[3] RoboDK. Franka emika panda robot. [Accessed: 2025-12-01]. [Online]. Available: https://robodk.com

[4] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno, "2d euclidean distance transform algorithms: A comparative survey," *ACM Comput. Surv.*, vol. 40, no. 1, Feb. 2008. [Online]. Available: https://doi.org/10.1145/1322432.1322434

[5] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[6] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[7] H. Kuroki and T. Baba, "Calligraphy z: A fabricatable pen plotter for handwritten strokes with z-axis pen pressure," in *Adjunct Proc. 35th Annu. ACM Symp. User Interface Software and Technology (UIST)*, Bend, OR, USA, 2022.

[8] J. Lee, M. Seo, A. Bylard, R. Sun, and L. Sentis, "Real-time model predictive control for industrial manipulators with singularity-tolerant hierarchical task control," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, London, United Kingdom, 2023, pp. 12 282–12 288.

[9] Z. Erickson, "Mengine: A manipulation and robotics simulation framework," https://github.com/Zackory/mengine, 2025, accessed: 2025-10-12.

[10] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 658–666.