Robot Drawing on a Moving Paperboard

Chonghao Cheng[1], Tiancheng Li[1], Jon Woolfrey[2], Liang Zhao[3], Shoudong Huang[1]
[1] Robotics Institute, University of Techonology Sydney, Chonghao.Cheng@student.uts.edu.au
[2] School of Electronic and Electrical Engineering, University of Leeds
[3] School of Informatics, The University of Edinburgh

Abstract

Robustness to uncertainties is crucial to collaborative robot control, particularly when performing tasks under unpredictable external influences. One important class of challenging problems is controlling the robot to perform a specific task on a target that undergoes random movements. To address this challenge, we consider the problem of using a robot manipulator to draw a given shape on a paperboard that can have unpredictable small movements. An electromagnetic sensor is used to monitor the pose of the paperboard. We design an MPC (Model Predictive Control) controller and compared it with a few different feedback control strategies for the robot to draw the required shape on the paperboard. Simulations and physical experiments using a Dobot demonstrate that using the electromagnetic sensor information as feedback combined with different control strategies can allow for accurate drawing. A video of our experiments is available at `https://youtu.be/Wmj0GU5CYH4?si=cgnN9D3VRr0ZLRlK`

1 Introduction

Collaborative robot control has been a critical research topic in the field of robotics and has found widespread applications in both industry and education for a long time. However, in real-world scenarios, the working environment of robotic arms is not always as controlled or interference-free as a factory setting. In such environments, the uncertainties can significantly impact the accuracy of robotic arm operations.

How can a robotic arm operate accurately under uncertainties in the environment? The environment uncertainties can be broadly categorized into three scenarios: (1) the platform on which the robotic arm is mounted can move, (2) the object that the robotic arm is manipulating can move, and (3) both the platforms where the robotic arm is installed and the object under manipulation can move. In the cases of factory assembly lines or surgical robot operations, the majority of cases fall under the second scenario, because the robotic arms are typically considered to be mounted on stable surfaces.

In this paper, we will focus on the second scenario, where the robotic arm is tasked with operating on a moving object. Existing research on this scenario largely assumes that the object being manipulated in a stable environment or under predictable disturbance[Lin and Görges, 2020; Li *et al.*, 2023a; Binder *et al.*, 2019].If the motion of the object or the platform on which the object rests is random and unpredictable, many methods may fail to achieve their claimed level of precision.

Moreover, most studies on robotic stability focus on tasks like object grasping, which are completed in a short, instantaneous action, without requiring sustained operations[Yin *et al.*, 2021; Zimmermann *et al.*, 2021]. In contrast, some robotic tasks involve continuous operations along a pre-determined trajectory, which imposes much higher demands on the robot's stability and robustness. The quality of the task execution is required to be extremely precise, as even a slight deviation can have a significant impact. For example, when an orthopedic surgical robot performs surgery automatically, the patient may experience slight vibrations or motions during the surgical process which affects the accuracy of the robot's operation [Walker *et al.*, 2022]. Therefore, stricter requirements for precision and reliability under such uncertainties are needed.

This paper considers the robot drawing problem, which requires accuracy over a sustained period of time. We assume the paperboard on which the robot is drawing can have unpredictable small movements. In order to achieve high accuracy in the drawing, we mount an electromagnetic sensor on the paperboard to track the pose of the paperboard and use this information to guide the control of the robot arm. We proposed a control method based on MPC and compared it with three different feedback control strategies in simulation, and conduct physi-

cal experiment using electromagnetic sensor. Our results show that this electromagnetic sensor based on our control strategy can achieve good accuracy.

This paper is organized as follows. Section 2 introduces the related works on robot manipulations under uncertainties; Section 3 presents the problem formulation; Section 4 talks about the our control methods and compared methods; Section 5 shows the simulation results and physical experiment setup and the results; Finally, Section 6 presents the conclusions and some topics for future work.

2 Related Works

The collaborative robots are composed of links connected by joints into a kinematic chain, which are widely applied to many areas [Spong *et al.*, 2020]. PID control is the most commonly used feedback control algorithm that combines proportional, integral, and derivative actions to maintain a desired setpoint by adjusting the control input based on current and past errors [Borase *et al.*, 2021]. Some methods control the robot by calculating the torque required to compensate for dynamic changes [Nguyen-Tuong *et al.*, 2008]. Another classical control method is Passivity-Based Control (PBC), which is a control strategy grounded in the concept of passivity from systems theory. It is particularly useful for managing dynamic systems, such as robotic and mechanical systems, where energy considerations are crucial [Ortega *et al.*, 2017].

However, most of the methods above mainly focus on the deterministic case or stable environment [Cui *et al.*, 2012]. With the development of robotics, it is natural to expect manipulators to be able to work in dynamic environments, such as underwater or on transportation equipments. Visual servoing control is a classical method for controlling a robot against a dynamic target, which utilizes a camera and image processing techniques to track the target position in real time and adjust the trajectory of the robotic arm based on the visual information [Chaumette and Hutchinson, 2006]. Adaptive control can adjust the control parameters in real time based on the dynamic characteristics of the target to accommodate its movement changes and is also suitable for uncertain or time-varying systems [Åström, 1995]. Model Predictive Control (MPC) is one of the most popular methods to handle those complex scenarios. It employs a system model to predict future states and optimize control inputs for precise tracking of moving targets [Kouvaritakis and Cannon, 2016]. Disturbance observer based control (DOB) methods have also garnered considerable attention in control theory research. They are designed to handle uncertainties and disturbances effectively and have good performance in real time, but they depend heavily on the accuracy of the system model and finding the optimal parameters for disturbance observers requires expertise and can be time-consuming [Chen *et al.*, 2015]. In recent years, reinforcement learning based control methods turn into mainstream. Reinforcement learning is a subfield of machine learning, concerned with how to find an optimal behavior strategy to maximize the outcome through trial and error dynamically and autonomously [Kalashnikov *et al.*, 2018].

With respect to robot drawing tasks, [Kemp and Edsinger, 2006] control the movement of the pen tip to draw marks by having the robot recognize the tip of the tool and identify it as an image area. [Yussof *et al.*, 2005] combine linear and curved trajectories in different ways to produce distinct characters. Aonther framework is proposed to combine trajectories to produce distinct symbols rather than just simple letters [Zhang and Weng, 2007]. In recent years, more computer vision techniques are being used. [Pichkalev *et al.*, 2019] describes the algorithm of points extraction from the picture processed by a canny edge detector that will be turned in the curves using the cubic spline interpolation, and use KUKA robot take the commands to drawing the result image. [Tsai *et al.*, 2021] use Dobot Magician and the Raspberry Pi development platform to integrate image processing and robot-arm drawing. However, those methods are mainly focus on how to use robot to draw on a stationary paper or board. In this paper, we consider the problem of a robot drawing on a paperboard that can have a small movement, especially when that movement is unpredictable. Also, instead of using vision for the feedback, we use an electromagnetic sensor to monitor the pose of the paper.

3 Problem Formulation

3.1 Problem Statement

In this paper, we consider the problem of using a robot to draw specified shapes on a paperboard which can move randomly. To simplify the problem, we suppose that the paperboard can only move in a 2D plane such as on a table surface. A marker is attached to the robot end-effector and can be used to draw on the paper. In order to achieve accurate drawing, we mount an electromagnetic (EM) sensor on the paperboard which can track the pose of the paper. Figure 1 shows an example of the experiment setup.

3.2 System model

In our control problem, the world coordinate system is defined by the robot base $\{B\}$. The states of the system consist of the robot end-effector position and the pose of the paperboard.

For the Dobot robot used in our experiment, it can only move in 4 degrees of freedom (DOF) including a 3D xyz position and a 1D orientation along z axis. During

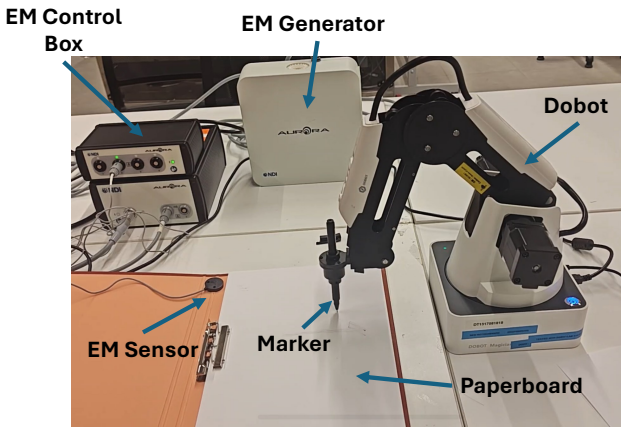


Figure 1: Physical experiment setup: Using the Dobot to draw a specific shape on paperboard that is moving randomly. The electromagnetic (EM) system (including generator, control box and sensor) is used to monitor the motion of the paperboard.

the drawing, we can restrict the robot operation to a fixed height (z is constant). In addition, since a marker is used for the drawing, the robot end-effector orientation is not so important and we mainly care about the end-effector xy position. At time step k, the end-effector xy position (in coordinate frame $\{B\}$) is denoted as

$$\mathbf{x}_R(k) = [x_r(k),\ y_r(k)]^{\mathrm{T}}, \tag{1}$$

and $\mathbf{x}_R(k)$ is **the first part of our state vector**.

Since the robot can move to any position within its work envelope, we simplify the **control input** as the change in the xy position, in coordinate frame $\{B\}$,

$$\mathbf{u}(k) = [x_u(k),\ y_u(k)]^{\mathrm{T}}. \tag{2}$$

Thus, the state transition model of $\mathbf{x}_R(k)$ is simply as

$$\mathbf{x}_R(k+1) = \mathbf{x}_R(k) + \mathbf{u}(k). \tag{3}$$

For the drawing, we would like the robot end-effector position to follow a pre-determined trajectory on the paperboard. Since the paper is moving, the target trajectory should be defined in the coordinate system defined by the paper, $\{P\}$. We denote the k-th point in the target trajectory as

$$\mathbf{L}^{\{P\}}(k) = (x_t^{\{P\}}(k), y_t^{\{P\}}(k)). \tag{4}$$

For accurate drawing, the relation between the two coordinate systems, $\{P\}$ and $\{B\}$, is critical. We use $T_{\{B\}\{P\}}(k) \in SE(2)$ to denote the relative coordinate transformation between the paper frame and the robot base frame at time k, and denote

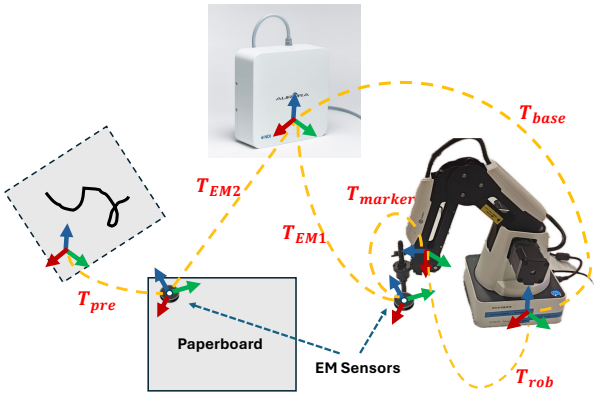


Figure 2: Chain of rigid transformation of the system

$$\mathbf{x}_P(k) = \log(T_{\{B\}\{P\}}(k)) \in se(2) \tag{5}$$

as **the second part of our state vector**. Since we assume the paper can have unpredictable small motions, the state transition model for $\mathbf{x}_P(k)$ can be described as

$$\mathbf{x}_P(k+1) = \mathbf{x}_P(k) + \mathbf{w}(k). \tag{6}$$

where $\mathbf{w}(k) \in se(2)$ represents the unpredictable motion from step k to step $k+1$.

Since the robot end-effector position at any time can be obtained directly, we can assume $\mathbf{x}_R(k)$ is fully observable without any observation noise, this is our **first observation** equation

$$\mathbf{z}_R(k) = \mathbf{x}_R(k). \tag{7}$$

We use an EM sensor mounted on the paperboard to track the paper position and orientation. The EM reading gives the pose of the EM sensor in the EM generator coordinate frame. After the calibration the EM sensor reading (T_{EM2} in Figure 2) can be regarded as another **observation** which is related to $\mathbf{x}_P(k)$, as follows

$$\begin{aligned} \mathbf{z}_P(k) &= \log(T_{base}^{-1} \cdot T_{\{B\}\{P\}}(k) \cdot T_{pre}^{-1}) + v_R(k) \\ &= \log(T_{base}^{-1} \cdot \exp^{\mathbf{x}_P(k)} \cdot T_{pre}^{-1}) + v_R(k) \end{aligned} \tag{8}$$

where $v_R(k)$ represents the EM sensor reading noise. The details of the calibration process can be found in Section 5.2.

Our control problem is to design control input $\mathbf{u}(k)$ *based on the target trajectory* $\mathbf{L}^{\{P\}}(j),\ j \geq 0$ *and the observations* $\mathbf{z}_R(i), \mathbf{z}_P(i),\ i \leq k$, *such that the robot end effector position* $\mathbf{x}_P(k)$ *follows the target trajectory on the paper as accurately as possible.*

4 Control Methods

The control objective to make the robot end-effector follow the target trajectory on the paper can be described

in different formats, and solving the corresponding control problem exactly can be very challenging. In this section, we will introduce our designed MPC control method to solve our problem and compared it with three traditional control methods. The common idea of these three methods is to ignore the EM sensor reading noise in (8) such that we can approximate $\mathbf{x}_P(k)$ as a function of $\mathbf{z}_P(k)$,

$$\mathbf{x}_P(k) \approx \tilde{\mathbf{x}}_P(k) = \log(T_{base} \cdot \exp^{\mathbf{z}_P(k)} \cdot T_{pre}). \quad (9)$$

4.1 State equation

The motion of a rigid body, such as the platform for a mobile manipulator, can be modeled as a linear time-invariant system with affine disturbance[Woolfrey *et al.*, 2021], and using $\tilde{\mathbf{x}}_P(k)$, we can transfer the target trajectory in the paper frame $\{P\}$ into a trajectory in the base frame $\{B\}$. This trajectory in the base frame can then be used to design the control input $\mathbf{u}(k)$. Based on the definitions provided in Section 3.2, we can define the state equation of this system as:

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\mathbf{X}(k) + \mathbf{B_1}\mathbf{u}(k) + \mathbf{B_2}\mathbf{w}(k) \\ \mathbf{Z}(k) &= \mathbf{X}(k) \end{aligned} \quad (10)$$

where:

1. $\mathbf{X}(k) = [\mathbf{x}_R(k), \mathbf{x}_P(k)]^{\mathrm{T}} \in \mathbf{R}^{5\times 1}$ is the state vector represents the position and orientation in the counterclockwise direction of the paperboard in 2D space and end effector position;
2. $\mathbf{u}(k) = [x_u(k),\ y_u(k)]^{\mathrm{T}} \in \mathbf{R}^{2\times 1}$ is the control value add to end effector position;
3. $\mathbf{A} \in \mathbf{R}^{5\times 5}$ is the identity matrix represents the state matrix;
4. $\mathbf{B_1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$ is the input matrix to adjust the control value to the state.
5. $\mathbf{B_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$
6. $\mathbf{w}(k) \in \mathbf{R}^{3\times 1}$ represents the unpredictable motion;
7. $\mathbf{Z}(k) = [\mathbf{z}_R(k), \mathbf{z}_P(k)]^{\mathrm{T}} \in \mathbf{R}^{5\times 1}$is the observation vector of the system.

4.2 Compared control methods

We use target trajectory to compute difference between target value at $k+1$ step and current end-effector position at k step to get $\mathbf{e}_{position}$ as controller input below.

$$\mathbf{e}_{position}(k) = \begin{bmatrix} \tilde{x}_t^{\{B\}}(k+1) \\ \tilde{y}_t^{\{B\}}(k+1) \end{bmatrix} - \begin{bmatrix} x_r(k) \\ y_r(k) \end{bmatrix} \quad (11)$$

where $\left[\tilde{x}_t^{\{B\}}(k+1), \tilde{y}_t^{\{B\}}(k+1)\right]^{\mathrm{T}}$ is computed by $\left[x_t^{\{P\}}(k+1), y_t^{\{P\}}(k+1)\right]^{\mathrm{T}}$ and $\tilde{\mathbf{x}}_P(k)$.

The first control method is the most common position control, utilizing the proportional control component as follows.

$$\mathbf{u}_1(k) = (1 + K_p) \cdot e_{position}(k) \quad (12)$$

where the proportional gain K_p is a manually adjusted parameter.

In the second control method, we build upon the first approach by adding compensation for the difference in observed values $\mathbf{z}_P(k)$ and $\mathbf{z}_P(k-1)$. This difference is obtained by comparing the observations from the previous and current steps. And this difference can be seen as a predictive velocity of the paper motion and thus can be applied as the feedback term on current controller.

$$\mathbf{u}_2(k) = (1+K_p)\cdot e_{position}(k) + K_c \cdot \mathbf{R}(k-1)\mathbf{t}(k-1) \quad (13)$$

where K_c is another gain parameter, and $\mathbf{R}(k-1)$ and $\mathbf{t}(k-1)$ are the rotation part and the translation part of $\exp^{(\mathbf{z}_P(k)-\mathbf{z}_P(k-1))}$.

The third method adds integral component based on control method 1. Multiplying K_i with the cumulative sum of $\mathbf{e}_{position}$ from start to current step as the integral control term

$$\mathbf{u}_3(k) = (1+K_p)\cdot e_{position}(k) + K_i \cdot \sum_{l=1}^{k} e_{position}(l) \quad (14)$$

4.3 Our method

MPC (Model Predictive Control) is an advanced control method that optimizes control inputs by predicting a system's future behavior over a time horizon, using a mathematical model. At each step, MPC minimizes a cost function, often balancing tracking accuracy and control effort, while respecting constraints on inputs and states. In this paper, we define the MPC problem in a discrete state framework, predicting the system's state over N steps to optimize and obtain the most suitable control values. We assume our control method can compensates for the movement of the paper perfectly then we can get a new equation for optimization to substitute the state equation.

$$\begin{aligned} \bar{\mathbf{X}}(k+N) &= \mathbf{A}^N \bar{\mathbf{X}}(k) + \sum_{i=0}^{N-1} \mathbf{A}^i \mathbf{B_1} \mathbf{u}(k+i) \\ X_k &= \begin{bmatrix} \bar{\mathbf{X}}(k+1) \\ \bar{\mathbf{X}}(k+2) \\ \cdots \\ \bar{\mathbf{X}}(k+N) \end{bmatrix} = \tilde{\mathbf{A}} X_{k-1} + \tilde{\mathbf{B}} U(k) \end{aligned} \quad (15)$$

where

1. $X_k \in \mathbf{R}^{5N\times 1}$ includes all the states of current end-effector position and paper pose use for prediction in total N steps;
2. $\tilde{\mathbf{A}} = \left[\mathbf{A}, \mathbf{A}^2, \ldots, \mathbf{A}^n\right]^{\mathbf{T}} \in \mathbf{R}^{5N\times 5}$ includes all state matrix in N steps;
3. $\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} & 0 & ... & 0 \\ \mathbf{AB_1} & \mathbf{B_1} & ... & 0 \\ ... & ... & ... & ... \\ \mathbf{A}^{N-1}\mathbf{B_1} & \mathbf{A}^{N-2}\mathbf{B_1} & ... & \mathbf{B_1} \end{bmatrix} \in \mathbf{R}^{5N\times 5N}$;
4. $U_k = \left[\mathbf{u}(k), \mathbf{u}(k+1), ..., \mathbf{u}(k+N-1)\right]^{\mathbf{T}} \in \mathbf{R}^{2N\times 1}$ is a vector of the predicted control value in N steps;
5. N is the prediction horizon.

We propose an optimized error metric suitable for optimization defined as follows:

$$\mathbf{e}(k) = \left[\left(x_r(i) - x_t(i)^{\{B\}}\right), \left(y_r(i) - y_t(i)^{\{B\}}\right)\right]^{\mathbf{T}} \quad (16)$$

and we can get E_k to represent vector of the error value in N steps:

$$E_k = \left[\mathbf{e}(k), \mathbf{e}(k+1), \ldots, \mathbf{e}(k+N)\right]^{\mathbf{T}} \quad (17)$$

Then we can use a cost function for optimization:

$$J = E_k^{\mathbf{T}}\mathbf{Q}E_k \quad (18)$$

$\mathbf{Q}$ represents the weights for error, and $\mathbf{R}$ represents the weights for the inputs. Then the problem can be formulated as;

$$minJ(E_k, U_k) \quad (19)$$

Our objective is to minimize the cost function, and we use quadratic programming (QP) as the optimization method to achieve this. Quadratic programming solves optimization problems where the objective function is quadratic, enabling us to obtain a set of control values U_k that minimize the cost function, thereby achieving our control goal. The use of the Our control methods and compared methods in the robot drawing can be summarized in Algorithm 1.

Algorithm 1: Drawing robot control on Moving Paperboard

Data: target trajectory $\mathbf{L}^{\{P\}}$
Initialize end-effector position $\mathbf{x}_R(0)$;
while *k less than* K_{limit} **do**
  get $\mathbf{z}_R(k)$ from the robot
  read EM sensor to get $\mathbf{z}_P(k)$
  $e_{position}(k) \leftarrow NextTargetInBase(\mathbf{L}^{\{P\}}(k+1), \mathbf{z}_P(k))$;
  $u_1, u_2, u_3 \leftarrow CalculateControl(\mathbf{x}_R(k), e_{position}(k), \mathbf{z}_P(k))$;
  $u_{MPC} \leftarrow CalculateOptimization(minJ(E_k, U_k))$;
  $\mathbf{x}_R(k+1) \leftarrow Move(\mathbf{x}_R(k), u_{(1,2,3,MPC)})$;
end

5 Experiments

5.1 Simulation Experiments

Simulation experiments are performed to compare the different control methods used in three target trajectories with two different paper movement types.

Simulation setup

The three target trajectories on the paper are given using the following waypoints ($0 \le k \le 20$) as follows. The first target trajectory is a straight line

$$\mathbf{L}_1^{\{P\}}(k) = (x_t^{\{P\}}(k), y_t^{\{P\}}(k)) = (2k-10, 0). \quad (20)$$

The second target trajectory is a quadratic curve

$$\mathbf{L}_2^{\{P\}}(k) = (x_t^{\{P\}}(k), y_t^{\{P\}}(k)) = (2k-10, \frac{(2k-10)^2}{25} - 4). \quad (21)$$

and the third target trajectory is a sine wave

$$\mathbf{L}_3^{\{P\}}(k) = (x_t^{\{P\}}(k), y_t^{\{P\}}(k)) = (2k-10, \frac{sin(2k)}{2}). \quad (22)$$

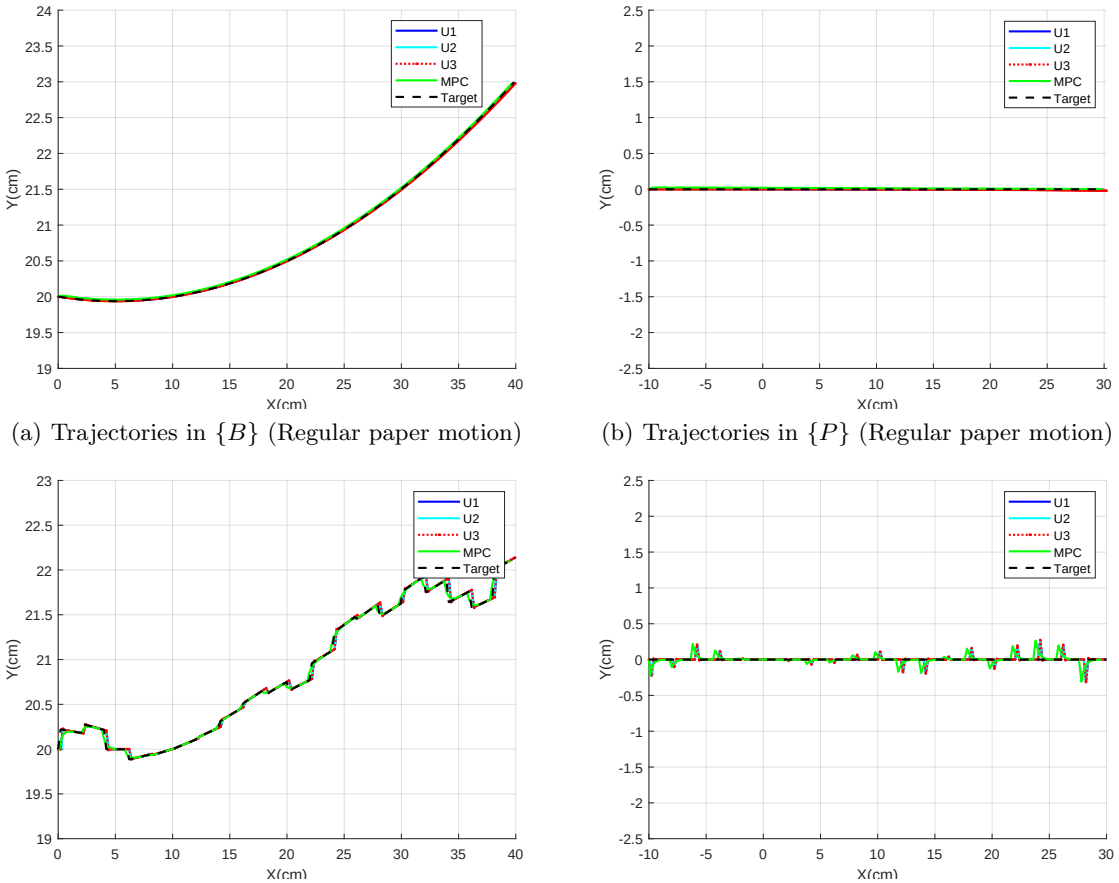
We assume that the origin of the paper frame coincides with the center of the paper, and the axis $X^{\{P\}}$ and $Y^{\{P\}}$ are set to be parallel to the robot's axis $X^{\{B\}}$ and $Y^{\{B\}}$ with the origin located at the robot coordinates frame $(10, 20)$. Then the initial position of the end-effector is set to $(x_r^{\{P\}}(0) = -10, y_r^{\{P\}}(0) = 0)$.

We consider a series of different paper movements. By changing the (x, y, θ) values of the paper's center in the robot coordinate frame, we simulated the EM sensor readings, also known as $\mathbf{x}_P(k)$ in equation (5).

We consider two types of movements in simulation experiment: random movements and regular periodic rotations. For random movements we give random value with a normal distribution in both (x, y, θ) every step, and for regular periodic movements we increase the θ with $2.5 \times 10^{-3} rad$ every step. To simulate the true EM sensor reading we also add random value with a normal distribution as noise.

Simulation results

We implemented and compared the three control methods discussed in Section 4. To visualize the drawing process, we recorded the end-effector positions during the drawing process and plotted it with the trajectory line in Fig. 3(a)(c). To conveniently demonstrate the performance of these different control methods, as well as to visually compare the differences between the lines drawn by the robot and the trajectory line, we also plotted the



(a) Trajectories in $\{B\}$ (Regular paper motion)

(b) Trajectories in $\{P\}$ (Regular paper motion)

(c) Trajectories in $\{B\}$ (Random paper motion)

(d) Trajectories in $\{P\}$ (Random paper motion)

Figure 3: Simulation experiment results for drawing the strainght line: the green line represents control method 1, the blue line represents control method 2, and the red line represents control method 3. The black dashed line indicates the target line. In subfigures (a) and (b), we simulated the paper being subjected to periodic rotational motions, while in subfigures (c) and (d), we simulated random motion of the paper. Subfigures (a) and (c) are presented in the robot base frame to show the actual motion trajectories, while subfigures (b) and (d) are shown in the paper frame to illustrate the deviation between the control algorithm's result and the target line.

trajectories in the paper coordinate frames shown in Fig. 3(b)(d). The black line is the target trajectory, in Fig. 3(a)(c) was projected to the robot base. The red line represents the result from control method u_1, the blue line is the result from control method u_2, and the green line is the result from control method u_3.

We also compared how the different disturbance affect the control process. In Fig. 3(a)(b) we set regular periodic rotations, and in Fig. 3(c)(d) random disturbances.

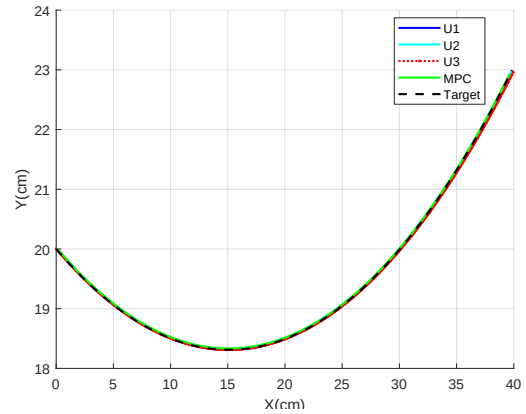
The results of drawing the quadratic curve and the sine wave are presented in Fig. 4 and Fig. 5.

A performance metric

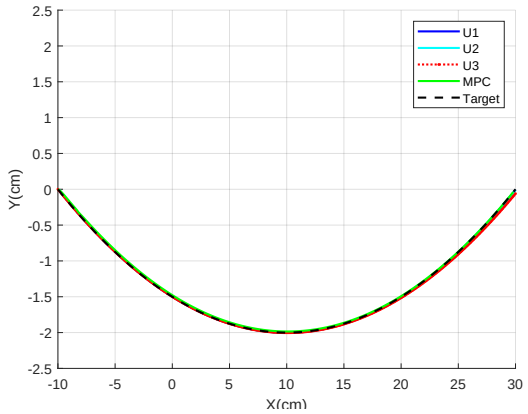
In order to evaluate the performance of different control methods quantitatively, we compute the area between the actual robot end-effector trajectory and the target trajectory. This can be computed relatively easily in the paper coordinate frame. Firstly, the robot end-effector position in each step is transferred into the paper coordinate frame using the ground truth paper poses, and we assume the movement trajectory between the robot's current position and its next position is a straight line, then we can get the robot movement trajectory during the drawing process. We calculated the integral of the difference between the two trajectories to obtain the area, which is denoted as E to evaluate the control performance.

We have calculated $\mathbf{e}$ when using different control methods with different target trajectories and conduct repeated experiments for three times and calculate their average value of $\mathbf{e}$, then recorded them in Table 1.

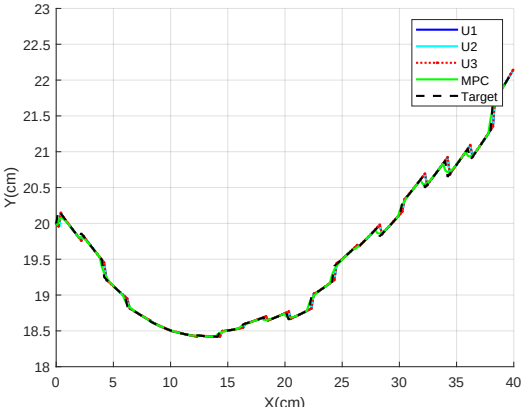
According to the results from Fig. 3 to Fig. 5 and Table 1, we can conclude that the our control method



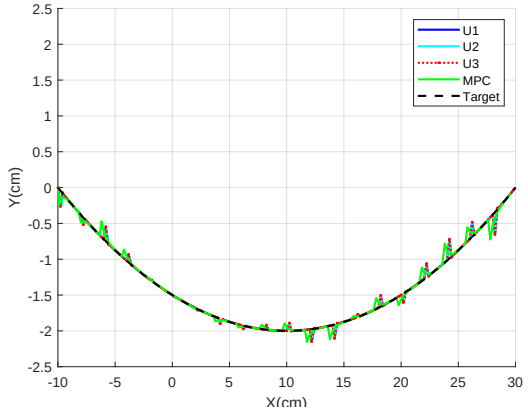
(a) Trajectories in $\{B\}$ (Regular paper motion)



(b) Trajectories in $\{P\}$ (Regular paper motion)



(c) Trajectories in $\{B\}$ (Random paper motion)



(d) Trajectories in $\{P\}$ (Random paper motion)

Figure 4: Simulation experiment results for drawing the quadratic curve.

Trajectory type	Disturbance Type	Control Method			
		U1	U2	U3	MPC
Line	Random	0.0098	0.0125	0.0570	0.0398
	Regular Rotation	0.0915	0.0934	0.0845	0.0960
Curve	Random	0.0110	0.0116	0.0617	0.0371
	Regular Rotation	0.0916	0.0915	0.3846	0.0960
Sine	Random	0.0125	0.0075	0.0680	0.0093
	Regular Rotation	0.0896	0.0971	0.0644	0.0954

Table 1: E (cm^2) of three control methods in simulation

have the best real-time performance and have an great performance in all cases for the considered simulation scenarios.

5.2 Physical Experiments

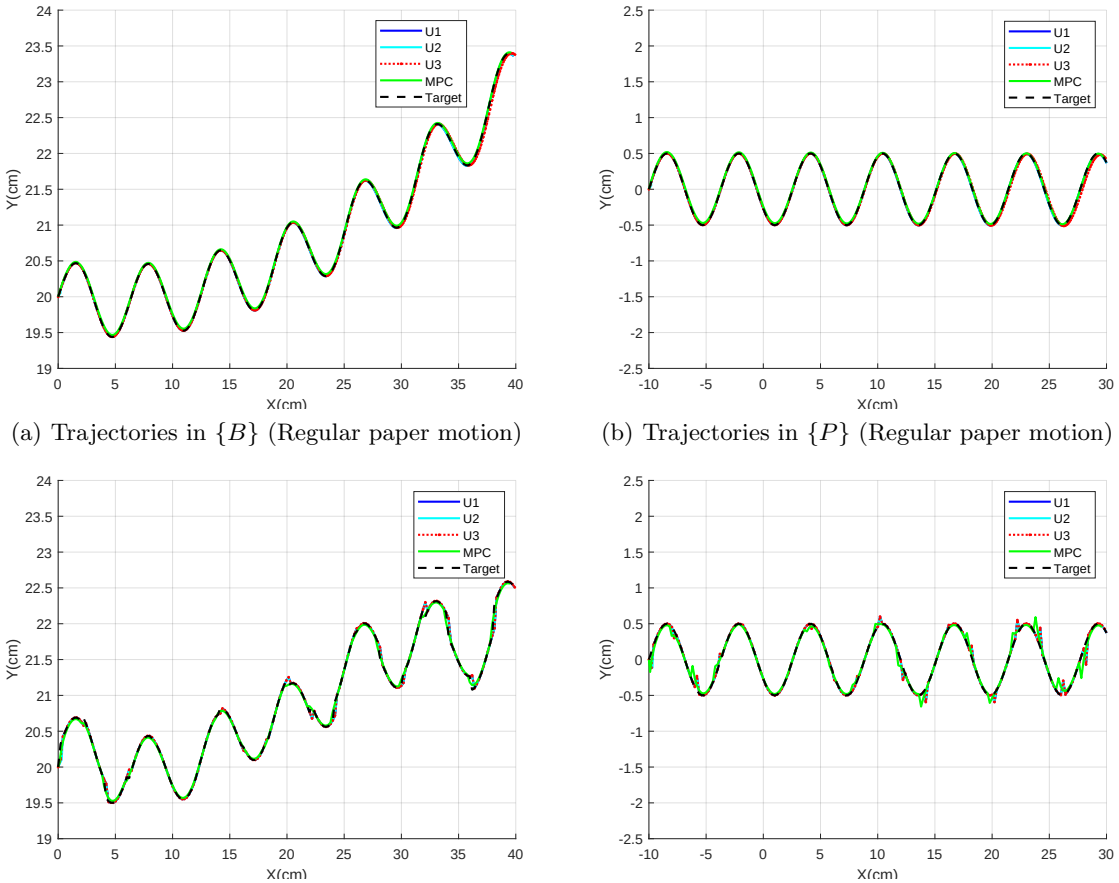
Experiment setup

The methods we compared in simulation are also tested in the physical experiments (Fig. 1). The Dobot Magician (Dobot Robotics, Shenzhen, China) is used to hold a marker to draw a straight line. Considering the challenging scenario that the paperboard can have random movement during the drawing, we attach an electromagnetic (EM) sensor on the paperboard to monitor its motion in real-time. The EM tracking system we use is the Aurora Electromagnetic Tracking (Northern Digital Inc., Canada) with planar field generator, which can provide six degrees of freedom measurements in 40 Hz [Sorriento *et al.*, 2019]. Its non-invasive nature and high sensitivity make it a critical technology for both industrial and medical applications [Li *et al.*, 2023b].

In order to compute the transformation between Dobot's base frame and the EM tracking frame (field generator), we attach another EM sensor to the marker (refer to Fig. 2) to perform a hand-eye calibration [Danilidis, 1999]. To solve the hand-eye calibration problem formulated as:

$$T_{rob} \cdot T_{marker} = T_{base} \cdot T_{EM1}, \tag{23}$$

the Dobot is moved arbitrarily to a sequence of poses



(a) Trajectories in $\{B\}$ (Regular paper motion)

(b) Trajectories in $\{P\}$ (Regular paper motion)

(c) Trajectories in $\{B\}$ (Random paper motion)

(d) Trajectories in $\{P\}$ (Random paper motion)

Figure 5: Simulation experiment results for drawing the sine wave trajectory.

and T_{rob}, forward kinematics of the robot and T_{EM1}, the transformation between the EM sensor and the field generator are stored in each pose (Fig. 2). The initial poses T_{base} can be written with respect to T_{EM1} as

$$T_{base} = T_{rob,0} \cdot T_{marker} \cdot T_{EM1,0}^{-1}. \tag{24}$$

Substituting (24) in (23) and rearranging it, gives a linear system of equations $AX = XB$ to solve for $X = T_{marker}$, where

$$\begin{aligned} A &= T_{rob,0}^{-1} \cdot T_{rob,k}, \\ B &= T_{EM1,0}^{-1} \cdot T_{EM1,k}, \end{aligned} \tag{25}$$

for $k = 0, \ldots, N$ poses. However, since the Dobot only covers one degree of freedom in rotation and three degrees of freedom in translation, it is not possible to simply apply the methods based on skew theory [Park and Martin, 1994] to solve the problem. Therefore, we apply a linear method based on dual quaternions [Ulrich *et al.*, 2014]. Substituting the calculated T_{marker} in (24) solves for T_{base}. Now the chain of transformations is closed.

To register the paperboard to pre-determined trajectory space, several distinctive landmarks are selected, i.e., the corners of the paperboard. The corresponding landmarks on the paperboard, are manually identified using an EM tracking probe. Then the rigid transformation T_{pre} can be calculated using these two sets of landmarks. Now the marker tip position x_{tip}^{marker} in the Dobot base frame can be found as:

$$x_{tip}^{marker} = T_{rob} \cdot T_{marker} \cdot T_{EM1}^{-1} \cdot T_{EM2} \cdot T_{pre} \cdot P_{start}, \tag{26}$$

where P_{start} is the start point of the drawing trajectory in the pre-determined frame.

Result

In the physical experiment, the we set the target tarjectory as a straight line and the start point of the marker tip on the paperboard can be found by (26). We used control method 2 to control Dobot. During the drawing, the paperboard attached with the EM sensor are slightly moved randomly. As shown in Fig. 6, the red line is the expected drawing line as the target trajectory and the black line is the actual line drawn by the

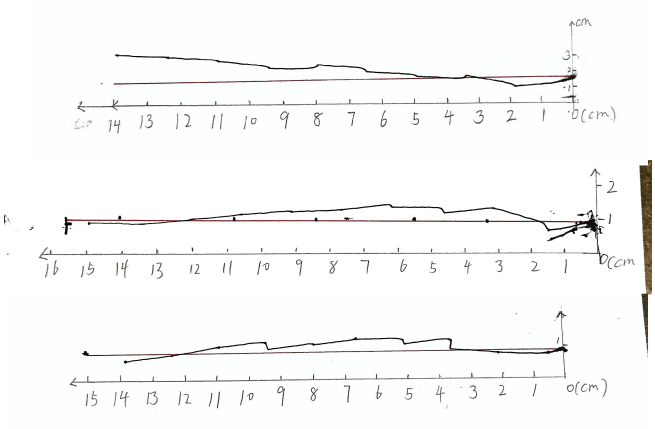


Figure 6: Three examples of robot drawing results from our physical experiments. Using control method 2 (velocity predictive term) to draw Line trajectory when paper is randomly moving.

Dobot. Due to the unpredictable movements of the paperboard, it is hard to match the actual drawing with the pre-determined line. But the control method effectively reduces the effect of cardboard movement, and the drawing results are not much off. A video of our experiments is available at Youtube `https://youtu.be/WmjOGU5CYH4?si=cgnN9D3VRrOZLRlK`

6 Conclusion and Future Work

This paper considers the problem of using a robot to draw lines on a paperboard that can move randomly. Electromagnetic (EM) sensors are used to track the paperboard motion and provide feedback for robot trajectory control. Simulation and practical experiments using a Dobot demonstrate the accuracy of our EM sensor based feedback control strategies.

In the future, we will first do more testing on robot performing more complicated drawings to evaluate the accuracy of the feedback control methods, and also develop more control algorithms taking into account the EM sensor observation errors. After that, we will extend our research into 3D where the paperboard can move arbitrarily in 3D space. For that case, we are planning to use a universal robot (such as UR10) which can achieve 3D orientation. Finally, we will investigate the surgical robot operation problem where a robot is required for performing accurate manipulations such as acetabular preparation for total hip replacement.

7 Acknowledgements

The authors would like to thank Dominik Slomma and Ziqi Wang for helping the Dobot programming.

References

[Åström, 1995] Karl Johan Åström. Adaptive control. In *Mathematical System Theory: The Influence of RE Kalman*, pages 437–450. Springer, 1995.

[Binder *et al.*, 2019] Benjamin Julian Tømte Binder, Tor Arne Johansen, and L Imsland. Improved predictions from measured disturbances in linear model predictive control. *Journal of Process Control*, 75:86–106, 2019.

[Borase *et al.*, 2021] Rakesh P Borase, DK Maghade, SY Sondkar, and SN Pawar. A review of pid control, tuning methods and applications. *International Journal of Dynamics and Control*, 9:818–827, 2021.

[Chaumette and Hutchinson, 2006] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[Chen *et al.*, 2015] Wen-Hua Chen, Jun Yang, Lei Guo, and Shihua Li. Disturbance-observer-based control and related methods—an overview. *IEEE Transactions on industrial electronics*, 63(2):1083–1095, 2015.

[Cui *et al.*, 2012] Ming-Yue Cui, Xue-Jun Xie, and Zhao-Jing Wu. Dynamics modeling and tracking control of robot manipulators in random vibration environment. *IEEE Transactions on Automatic Control*, 58(6):1540–1545, 2012.

[Daniilidis, 1999] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298, 1999.

[Kalashnikov *et al.*, 2018] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

[Kemp and Edsinger, 2006] Charles C Kemp and Aaron Edsinger. Robot manipulation of human tools: Autonomous detection and control of task relevant features. In *Proc. of the Fifth Intl. Conference on Development and Learning*, volume 42, 2006.

[Kouvaritakis and Cannon, 2016] Basil Kouvaritakis and Mark Cannon. Model predictive control. *Switzerland: Springer International Publishing*, 38:13–56, 2016.

[Li *et al.*, 2023a] Pengfei Li, Yu Kang, Tao Wang, and Yun-Bo Zhao. Disturbance prediction-based adaptive event-triggered model predictive control for perturbed nonlinear systems. *IEEE Transactions on Automatic Control*, 68(4):2422–2429, 2023.

[Li *et al.*, 2023b] Tiancheng Li, Yang Song, Peter Walker, Kai Pan, Victor A van de Graaf, Liang Zhao, and Shoudong Huang. A closed-form solution to electromagnetic sensor based intraoperative limb length measurement in total hip arthroplasty. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 365–375. Springer, 2023.

[Lin and Görges, 2020] Xiaohai Lin and Daniel Görges. Robust model predictive control of linear systems with predictable disturbance with application to multiobjective adaptive cruise control. *IEEE Transactions on Control Systems Technology*, 28(4):1460–1475, 2020.

[Nguyen-Tuong *et al.*, 2008] Duy Nguyen-Tuong, Matthias Seeger, and Jan Peters. Computed torque control with nonparametric regression models. In *2008 American Control Conference*, pages 212–217. IEEE, 2008.

[Ortega *et al.*, 2017] Romeo Ortega, Alejandro Donaire, and Jose Guadalupe Romero. Passivity-based control of mechanical systems. *Feedback Stabilization of Controlled Dynamical Systems: In Honor of Laurent Praly*, pages 167–199, 2017.

[Park and Martin, 1994] Frank C Park and Bryan J Martin. Robot sensor calibration: solving ax= xb on the euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5):717–721, 1994.

[Pichkalev *et al.*, 2019] Maxim Pichkalev, Roman Lavrenov, Ramil Safin, and Kuo-Hsien Hsia. Face drawing by kuka 6 axis robot manipulator. In *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, pages 709–714. IEEE, 2019.

[Sorriento *et al.*, 2019] Angela Sorriento, Maria Bianca Porfido, Stefano Mazzoleni, Giuseppe Calvosa, Miria Tenucci, Gastone Ciuti, and Paolo Dario. Optical and electromagnetic tracking systems for biomedical applications: A critical review on potentialities and limitations. *IEEE reviews in biomedical engineering*, 13:212–232, 2019.

[Spong *et al.*, 2020] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. John Wiley & Sons, 2020.

[Tsai *et al.*, 2021] Pu-Sheng Tsai, Ter-Feng Wu, Jen-Yang Chen, and Fu-Hsing Lee. Drawing system with dobot magician manipulator based on image processing. *Machines*, 9(12):302, 2021.

[Ulrich *et al.*, 2014] Markus Ulrich, Andreas Heider, and Carsten Steger. Hand-eye calibration of scara robots. *OGRW2014*, page 117, 2014.

[Walker *et al.*, 2022] Peter Walker, Tiancheng Li, Richardo Khonasty, KM Ponnanna, Alexander Kuo, Liang Zhao, and Shoudong Huang. Proof of concept study for using ur10 robot to help total hip replacement. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 18(2):e2359, 2022.

[Woolfrey *et al.*, 2021] Jon Woolfrey, Wenjie Lu, and Dikai Liu. Predictive End-Effector Control of Manipulators on Moving Platforms Under Disturbance. *IEEE Transactions on Robotics*, 37(6):2210–2217, December 2021.

[Yin *et al.*, 2021] Hang Yin, Anastasia Varava, and Danica Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54):eabd8803, 2021.

[Yussof *et al.*, 2005] Salman Yussof, Adzly Anuar, and Karina Fernandez. Algorithm for robot writing using character segmentation. In *Third International Conference on Information Technology and Applications (ICITA'05)*, volume 2, pages 21–24. IEEE, 2005.

[Zhang and Weng, 2007] Yilu Zhang and Juyang Weng. Task transfer by a developmental robot. *IEEE Transactions on Evolutionary Computation*, 11(2):226–248, 2007.

[Zimmermann *et al.*, 2021] Simon Zimmermann, Roi Poranne, and Stelian Coros. Go fetch!-dynamic grasps using boston dynamics spot with external robotic arm. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4488–4494. IEEE, 2021.