

Harnessing with Twisting: Single-Arm Deformable Linear Object Manipulation for Industrial Harnessing Task

Xiang Zhang¹, Hsien-Chung Lin², Yu Zhao², and Masayoshi Tomizuka¹

Abstract—Wire-harnessing tasks pose great challenges to be automated by the robot due to the complex dynamics and unpredictable behavior of the deformable wire. Traditional methods, often reliant on dual-robot arms or tactile sensing, face limitations in adaptability, cost, and scalability. This paper introduces a novel single-robot wire-harnessing pipeline that leverages a robot’s twisting motion to generate necessary wire tension for precise insertion into clamps, using only one robot arm with an integrated force/torque (F/T) sensor. Benefiting from this design, the single robot arm can efficiently apply tension for wire routing and insertion into clamps in a narrow space. Our approach is structured around four principal components: a Model Predictive Control (MPC) based on the Koopman operator for tension tracking and wire following, a motion planner for sequencing harnessing waypoints, a suite of insertion primitives for clamp engagement, and a fix-point switching mechanism for wire constraint updating. Evaluated on an industrial-level wire harnessing task, our method demonstrated superior performance and reliability over conventional approaches, efficiently handling both single and multiple wire configurations with high success rates.

I. INTRODUCTION

Recent advancements in robotics have seen significant progress in handling rigid objects [1]–[3], but extending these advanced dexterous skills to manipulate deformable objects like cables, wires, and clothing remains a challenge. This challenge arises primarily due to the complex behavior of deformable objects, which have a high-dimensional state space and nonlinear dynamics that complicate state estimation, path planning, and predictive modeling, making automation in real-world scenarios difficult. The complexity increases when these objects interact with their environment, as seen in tasks like wire harnessing [4]–[9].

Fig. 1 illustrates a typical wire harnessing setup, where a robot must shape a deformable linear object (DLO) - in this case, a wire - into a specific form and secure it with clamps. Unlike other deformable object manipulation tasks such as shape tracking [10], [11] or cloth folding [12], wire harnessing demands accurate state estimation to determine the wire’s shape and requires precise control over the force applied to tension the wire for insertion into clamps. Previous solutions have utilized dual-robot arms to create tension [5], [6], [8] or tactile sensors [4], [7] to gauge gripping and shearing forces, but these approaches have limitations. Dual-arm setups are costly and pose motion planning challenges, while tactile sensors can degrade over time [13], losing

effectiveness. Moreover, most prior research has focused on simplified scenarios with either loose clamps [7] that reduce the harnessing to a placing task or large gaps between clamps that make it easier for the robot to generate tension. Thus, the previous settings may not accurately reflect the complexities of real-world applications, such as the NIST board assembly challenge [14].

Our approach aims to address this complex task using a single robot arm equipped with force sensing. We leverage a simple yet effective strategy involving stretching and twisting motions. Stretching straightens the wire, simplifying state estimation and preparing it for insertion into a clamp, while twisting twines the wire to the robot gripper and adjusts the tension by changing the twisting angle. This combination allows a single arm to precisely position and tension the wire for insertion. Our proposed framework comprises four main components: 1) a Model Predictive Control (MPC) module that uses a Koopman-operator based dynamics function for motion planning and tension prediction, 2) a waypoint planning module for mapping out the path and avoiding collisions, 3) insertion primitives designed for securing the wire in clamps without losing tension, and 4) a fix-point switching mechanism for wire constraint updating. We evaluated our framework in the NIST board assembly challenge, a task that represents an industrial-level wire harnessing challenge. Our system demonstrated a high success rate in real-world wire-following trajectories, showcasing its potential to effectively tackle the demanding wire-harnessing task while managing tension forces.

II. RELATED WORK

A. Challenges of Deformable Manipulation

Manipulation of deformable objects has recently garnered significant interest in robotics research, with state estimation and dynamics modeling emerging as key challenges in automating such tasks.

State estimation is crucial for subsequent manipulation tasks. Traditional methods [15]–[17] often rely on color filtering to distinguish the object from its background, followed by non-rigid registration methods to derive a keypoint representation for planning and manipulation. This approach, however, depends heavily on manual tuning of filter parameters and is not robust to changes in task settings. Recent advancements have seen learning-based methods excel in identifying states of deformable objects. For instance, Jin et al. [8] introduced a deep neural network (DNN) that leverages real-world data to detect deformable linear objects

¹Mechanical Systems Control Lab, UC Berkeley, USA. {xiang_zhang_98, tomizuka}@berkeley.edu

²FANUC Advanced Research Laboratory, FANUC America Corporation, USA. {john.lin, Yu.Zhao}@fanucamerica.com

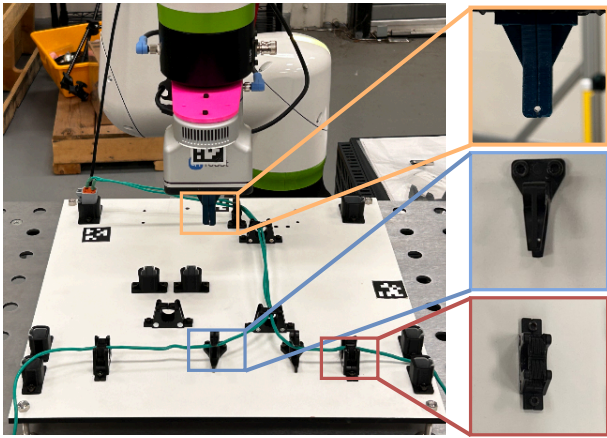


Fig. 1: NIST task board setup for wire harnessing task. We use a 3D-printed finger with grooves (orange box) to harness the wire into “C”-shaped (blue box) and “U”-shaped clamps (red box).

(DLOs) in an environment. Similarly, Yan et al. [18] employed a coarse-to-fine strategy using VGG feature maps to locate rope segments, enhancing resolution progressively. Furthermore, some end-to-end models [19] bypass the need for explicit state estimation by directly using visual inputs to predict manipulative actions or the dynamics of deformable objects. In our research, we simplify the state estimation challenge by straightening the wire, thereby facilitating easier handling and application of tension for the wire harnessing task.

Dynamics modeling of deformable objects is particularly difficult due to their infinite degrees of freedom. A common simplification involves representing these objects with a finite set of particles or connected rigid bodies [10], [16], though this often fails to accurately capture force interactions and can lead to non-intuitive model parameters [20]. Alternatively, learning-based methods have been developed to approximate the dynamics of deformable objects from data [10], [18], [19]. These models, depending on their state representations, can predict future states of the object, whether as key points or entire images, based on the robot’s actions. The choice of network, whether MLP, LSTM, or, as Wang et al. [10] have shown, graph neural networks (GNNs), significantly impacts performance. GNNs, in particular, are effective due to their structure, which inherently captures the dynamics of deformable objects. In our work, we simplify the system by applying tension to the wire, using the Koopman operator for dynamic modeling, which accelerates training and inference, enabling real-time optimal control.

B. Wire Harnessing Task

Wire harnessing presents a complex challenge in deformable object manipulation, requiring precise contact with the environment and deformation into specific shapes. The primary challenges are twofold: guiding the wire through a series of clamps and securing it within these clamps. Traditional approaches have explored dual-arm setups [5],

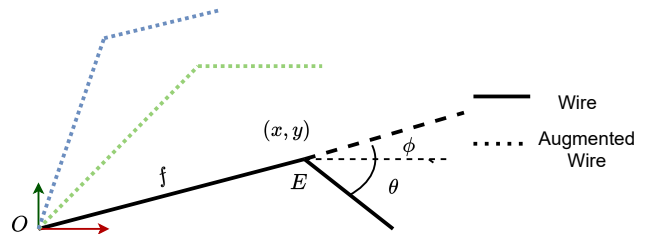


Fig. 2: State space of the wire harnessing task. O represents the fixed wire origin. E is the robot end-effector (gripper) to bend and twist the wire. The state space consists of (x, y, θ, f) denotes the robot position and rotation relative to O and the tension force. ϕ is the twisting angle between the wire and gripper.

[6], [8] or tactile sensors [4], [7]. In a dual-arm configuration, one arm holds the wire steady while the other applies tension for routing and insertion. Tactile setups rely on sensors to detect the wire’s pose and tension for guidance and insertion. However, these methods are either complex and costly or suffer from durability issues with tactile sensors. Our approach uses a single robot arm equipped with force-torque (FT) sensing, utilizing stretching and twisting motions to generate tension, simplifying state estimation and aiding in both routing and insertion tasks. The closest work to this paper is [5], whose authors adopted a dual-arm setup with FT sensing and occasionally uses single arm to stretch the wire for insertion. However, our setup is more challenging since the clamps are much smaller and tighter than their settings which requires the robot to generate enough tension in a limited space to finish the insertion. In the experiments, we demonstrate our approach can achieve faster force tracking performance with twisting motion and successes in this challenging real-world harnessing task.

III. PROBLEM FORMULATION

A. Wire Harnessing Task Set-up

The NIST Assembly Task Board 4 [14] serves as our testbed for the wire harnessing task, incorporating both connector assembly and wire harnessing tasks. Since our focus is not on the connector assembly, we proceed under the assumption that the connector is pre-attached to the board, and one end of the wire is already fixed to this connector. The robot’s objective is then to route the wire into a desired configuration by securing it within a series of clamps. The board features two clamp types: “C”-shaped clamps act as pivot points for wire routing, and “U”-shaped clamps secure the wire post-insertion. With the 3D geometry of the NIST task board known, we assume that each clamp’s relative position is predetermined, and the board’s location is identifiable via an AR tag, simplifying the vision-based clamp localization process. For environments where clamp detection is necessary, the segmentation and pose estimation method mentioned in [7] could be applied. Fig. 1 illustrates the task board and an example of the targeted wire configuration.

B. Robot Setup for Wire Harnessing

Robot Manipulator: Diverging from the dual-arm or tactile sensor-based approaches of previous studies, our method requires only a single robot arm equipped with force sensing capabilities. We employ a FANUC CRX-10iA collaborative robot for our experiments. Equipped with joint torque sensors, the robot can directly estimate the tension force during harnessing using the relationship $F = J^T \tau$, where $\tau \in \mathbb{R}^6$ represents the joint torque, $J \in \mathbb{R}^{6 \times 6}$ denotes the Jacobian matrix, and $F \in \mathbb{R}^6$ is the external wrench. For other robots without this capability, an external F/T sensor on the robot wrist can be utilized to directly measure the tension force F .

A RealSense L515 camera attached to the end-effector aids in acquiring the task board’s pose and generating the task board’s point cloud for initial wire grasping pose estimation.

Gripper Design: The standard parallel gripper, while commonly used, proves unsuitable for tasks requiring wire stretching or following due to the unpredictability of in-hand wire pose and the risk of wire slippage. As shown in Fig. 1, to overcome this, we redesigned the gripper to better suit our needs, incorporating a thin finger design with grooves that allow the wire to pass through, thereby securing it during harnessing and enhancing task robustness. The robot manipulates wire tension by actively twisting it, applying rotational motion to increase friction.

C. State and Action Spaces Design

State space: The wire’s state in the harnessing task can exhibit hybrid dynamics: either slack, exhibiting no tension and assuming random shapes, or stretched, forming a straight line under tension. Given that one wire end is always fixed (either to the connector or by clamps), and assuming pre-grasp by the robot in a planner stretching scenario, we consider a simplified 2D case as shown in Fig. 2, where the origin O is the fix-point and E is the robot end-effector, a gripper, twisting the wire (solid line). We define the wire state as $s = (x, y, \theta, f)^T$, where x, y, θ represent the robot gripper’s relative position to the fix-point O and Z axis rotation, respectively, and $f = \|F\|_2$ indicates the tension force’s 2-norm. This 2D case simplification omits Z direction motion, reducing system complexity. Additionally, since the tension force direction aligns with the robot’s pose, only the force magnitude is considered. Since the wire will be twisted and stretched throughout the harnessing process, this setup directly utilizes the robot gripper’s position to monitor the wire’s state.

Action space: The robot’s actions in wire harnessing combine stretching and twisting motions. Specifically, the control command u includes translational incremental motion $(\Delta x, \Delta y)^T$ for stretching and rotational incremental motion $\Delta \theta$ for twisting. Each command will be executed for 0.5 s by the robot. Throughout the task, the robot adjusts its actions to stretch and twist the wire, achieving the necessary tension for task completion.

D. Basics of Koopman Operator

Consider a discrete nonlinear dynamics system $s_{t+1} = F(s_t)$, where $s_t \in \mathbb{R}^n$, the Koopman operator: $\mathcal{K} : \mathcal{F} \rightarrow \mathcal{F}$ is an infinite linear transformation that evolves every observables $g : \mathbb{R}^n \rightarrow \mathbb{R}$ belonging to \mathcal{F} [21]:

$$(\mathcal{K}g)(s_t) = g(F(s_t)) = g(s_{t+1}) \quad (1)$$

In practice, we use a finite approximation of the infinite linear transformation with Koopman matrix $K \in \mathbb{R}^{m \times m}$ and use a lift function $g(x_t) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ to represent a set of base observation functions (g_1, \dots, g_m) . While the original Koopman operator does not consider the control term $u_t \in \mathbb{R}^l$, previous works [22]–[24] have shown it can be linearly combined with the Koopman observation space resulting in the form of:

$$g(s_{t+1}) = Kg(s_t) + Lu_t \quad (2)$$

where $L \in \mathbb{R}^{m \times l}$ is the control matrix.

IV. PROPOSED APPROACH

Our key idea employs stretching and twisting motions to generate the tension required for wire harnessing with a single arm. This approach consists of four main components: 1) Wire following with force tracking control, which plans stretching and twisting motions to guide the wire to the intended pose while maintaining tension; 2) Harnessing motion planning, which sequences the waypoints to complete the harnessing task; 3) Clamp insertion primitives, which use the wire’s tension for precise alignment with “U”-shaped clamps; and 4) a fix-point switching mechanism that updates the current wire constraints. The whole pipeline is depicted in Fig. 3(a).

A. Wire Following with Force Tracking Control

Achieving the right wire tension is crucial, as excessive force can damage the wire or connector, while a slack wire with insufficient tension makes clamp insertion challenging. The complexity lies in the wire’s nonlinear dynamics and the difficulty of determining optimal control commands. We address this by leveraging the Koopman operator for a linear dynamic model in the lift space, using real-world wire following trajectories. An MPC module then calculates the optimal robot movements for tension force tracking and waypoint navigation.

Koopman dynamics fitting: To fit the nonlinear wire dynamics, we define the lifting function $g(s_t)$ as follows:

$$g(s_t) = (s_t, \phi_t, z(s_t))^T = (x_t, y_t, \theta_t, f_t, \phi_t, z(s_t))^T \quad (3)$$

, where ϕ_t represents the twisting angle between wire and gripper at time t as shown in Fig. 2, and $z(s_t)$ includes the 2-order polynomial of (s_t, ϕ_t) . We chose this setup for its simplicity and adequacy in representing wire dynamics. We offline collected 40 real-world wire-following trajectories $[\tau_i = (s_0, u_0, \dots, s_T, u_T), i = 1, \dots, 40]$ as the dataset using scripted twisting and stretching motion with random initial

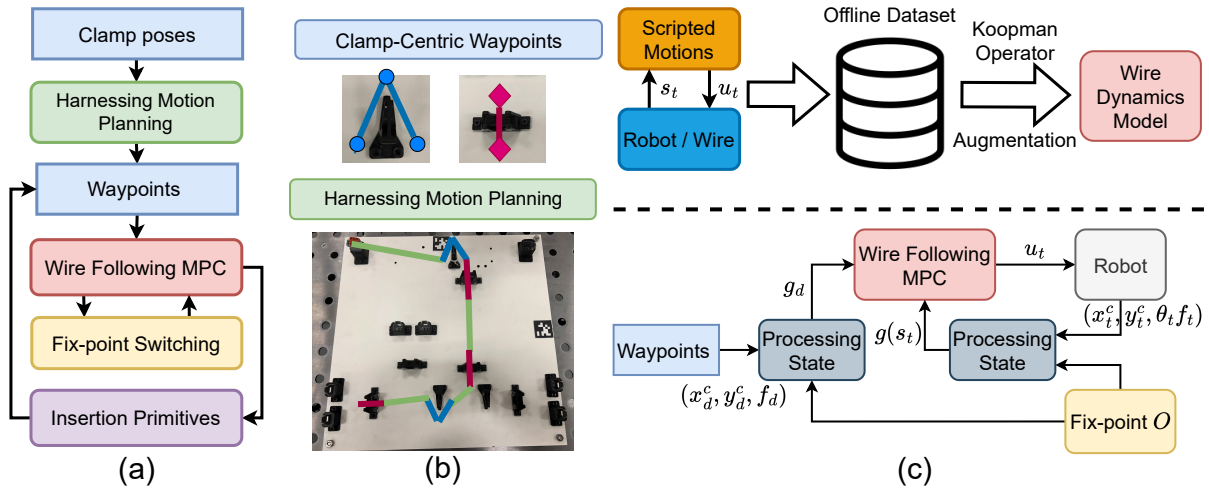


Fig. 3: The overview of our proposed approach: a) the pipeline for wire-harnessing, b) harnessing motion planner that sequences and merges a set of clamp-centric waypoints for each of the clamps, c) Top: We use pre-scripted motions to collect real-world wire following data, then augment by 10 times for Koopman operator fitting. Bottom: We first process the waypoints and robot state in the Cartesian space to the fix-point frame and use Koopman lift function to obtain the wire state. Then we utilize MPC to infer the optimal control command.

states. To fit the model, our objective is to minimize one-step prediction loss:

$$L = \sum_{i=1}^{N-1} \left\| g(s_{i+1}) - [K, L] \begin{bmatrix} g(s_i) \\ u_i \end{bmatrix} \right\|_2^2 \quad (4)$$

To simplify, we denote $\begin{bmatrix} g(s_i) \\ u_i \end{bmatrix}$ as $g'(s_i, u_i)$. Then, the Koopman and control matrix can be obtained by using the closed-form solution:

$$[K, L] = PG^\dagger \quad (5)$$

$$P = \frac{1}{N-1} \sum_{i=1}^{N-1} g(s_{i+1})g'(s_i)^T \quad (6)$$

$$G = \frac{1}{N-1} \sum_{i=1}^{N-1} g'(s_i)g'(s_i)^T \quad (7)$$

where \dagger is the Moore–Penrose pseudoinverse.

Geometric inspired data augmentation: Since we only have a limited number of training data, we also augmented them with geometric heuristics to improve the model performance. As shown in Fig. 2, our intuition is that the tension force will be identical if we rotate the stretching trajectory by an angle ψ . In practice, we augment each real-world trajectory by 10 times with a rotation angle from $[-\pi, \pi]$ with an interval of $\frac{1}{5}\pi$.

MPC setup: We utilize MPC to infer the optimal control command by solving the following optimization:

$$\begin{aligned} \min_{u_i, \dots, u_{i+H}} & \sum_{t=i}^H (g(s_t) - g_d)^T Q (g(s_t) - g_d) + u_t^T R u_t \\ \text{s.t.} & g(s_{t+1}) = K g(s_t) + L u_t \\ & b_l \leq A g(s_{t+1}) \leq b_u \\ & c_l \leq u_t \leq c_u, \forall t \in [i, i+H] \end{aligned} \quad (8)$$

where $g_d = (x_d, y_d, 0, f_d, 0, \dots, 0)^T$ represents the desired waypoint (x_d, y_d) and desired force f_d . $Q = \text{diag}(10, 10, 0, 1, 0, \dots, 0)$ focuses on penalizing only position and force tracking errors and ignores the rotation term, $R = \text{diag}(0.1, 0.1, 0.1)$ regulates control effort, $A = [\mathbb{I}^4, 0]$ is the selection matrix to extract the original state and $[b_l, b_u], [c_l, c_u]$ are lower and higher bounds for the state and control, respectively. We keep solving this problem for each step and only execute u_i for each time.

One thing to note is that the origin of the state is located at the fix-point but the way-points and the robot poses are planned or measured in the world frame. Therefore, at each control loop, we process the Cartesian space waypoint (x_d^c, y_d^c, f_d) , and robot state $(x_t^c, y_t^c, \theta_t, f_t)$ to the fix-point frame and then project them to the lift space to run the MPC.

B. Harnessing Motion Planning

Given a sequence of clamps for wire harnessing, we propose a clamp-centric waypoint planning algorithm. As shown in Fig. 3(b), for “U”-shape clamps, there are two waypoints for the robot to pull the wire on the top of the clamp and secure it with the insertion primitive. “C”-shape clamps have two side-waypoints and one tip-waypoint to route the wire around the clamp. The overall planning consists of two steps. In the first step, starting from the initial fix-point, we add the waypoints of the next clamp

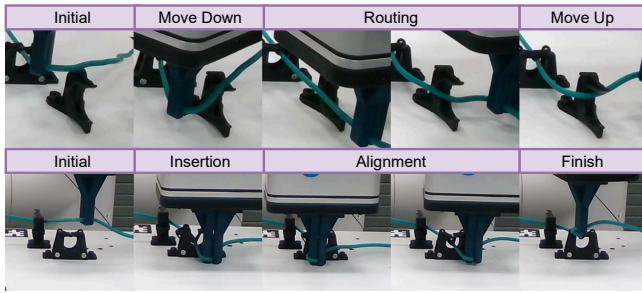


Fig. 4: Insertion primitives for “C”-shaped clamps (top) and “U”-shaped clamps

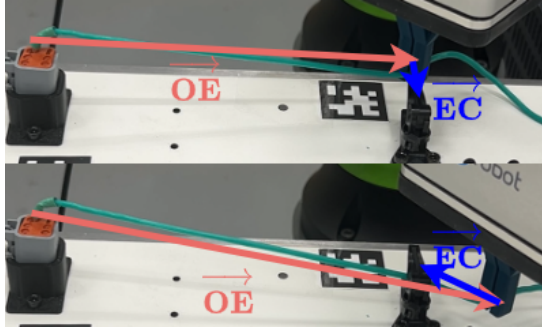


Fig. 5: Fix-point switching mechanism.

based on their distance to the most current waypoint. For “C”-shape clamps, the tip-waypoint must be in the middle of two side-waypoints. After obtaining a set of waypoints to follow, step two combines the waypoints that are within 20 mm distance to their mean to improve the efficiency of the planned path. The desired force f_d is assigned to 7 N for “C”-shaped clamp waypoints for routing and 10 N for “U”-shaped clamp waypoints since the insertion requires a greater tension.

C. Clamp Insertion Primitives

When the robot stretches the wire to the desired waypoint, we utilize designed insertion primitives to align it to the clamps. Fig. 4 depicts the snapshots of the primitive and their details are explained below:

“C”-shaped clamps: For the first side-waypoint, the robot moves down by 30 mm to lower its position for routing. After reaching the second side-waypoint, the robot moves up again for 30 mm to avoid collision with the board and finishes the routing.

“U”-shaped clamps: After reaching the second side-waypoint, the robot first goes down 30 mm for insertion while keeping twisting to maintain the wire tension. Then the robot moves along the edge of the clamp in two directions for 20 mm to further secure the wire. Finally, the robot moves up 30 mm to finish the insertion.

D. Fix-point Switching

Upon successful wire alignment, we update the fix-point, which is the origin of the state. Fig. 5 depicts the routing process for a “C”-shaped clamp, where O is the fix-point, E

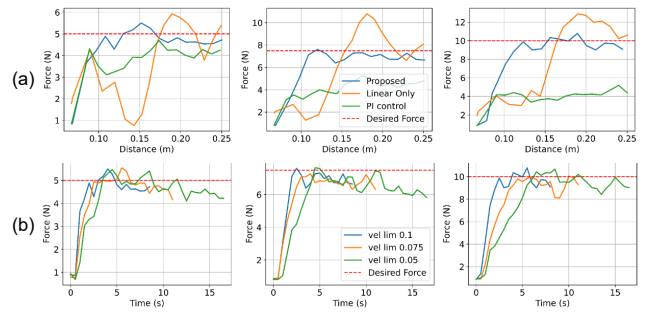


Fig. 6: Tension tracking comparison of a) Proposed approach against baselines and b) Effect of different tracking velocity limits

is the gripper, and C stands for the clamp. For “C”-shaped clamps, the wire’s binary position relative to the clamp is determined by $\overline{OE} \times \overline{EC} \cdot (0, 0, 1)^T$, which indicates the wire is on the left or right-hand side of the clamp. If $\|OE\| > \|OC\|$ which means the wire is stretched long enough to route on the clamp and the binary wire position changes the sign, we then switch the fix-point O to the the clamp E . For “U”-shaped clamps, we change the fix-point when the insertion primitive is executed.

V. EXPERIMENTS

In this section, we evaluate the wire-harnessing pipeline proposed in Section IV on the NIST assembly task board. To assess the effectiveness of the twisting motion in generating cable tension and the necessity of a Koopman operator-based dynamics model, we compare our method against two baselines: **No-twisting**, which adjusts stretching velocity using a PI controller to generate wire tension, and **Linear dynamics**, which employs a linear dynamics model $s_{t+1} = As_t + Bu_t$ with the same MPC setup for optimal robot motion determination. Specifically, we aim to address three questions related to wire-harnessing performance: 1) Can our Koopman-based MPC generate the desired wire tension? 2) Is the proposed pipeline capable of completing the complex wire-harnessing task? and 3) Can our method be extended to handle multiple wires?

A. Tension Tracking Performance

To address the first question, we examined our Koopman-based MPC’s tension-tracking capability. In our tests, we presuppose the robot is grasping the wire, and then stretching it by 250 mm in the Y direction to sustain tension forces of [5, 7.5, 10] N. Initially, we set the twisting angle ϕ to 0° and capped the robot’s maximum velocity at 0.1m/s and $6^\circ/s$ for translational and rotational movements, respectively. The force-tracking outcomes are showcased in Fig. 6(a). Our findings indicate that the Koopman-based MPC proficiently maintains the desired wire tension, unlike the No-twisting benchmark, which caps at a 4 N tension force since the robot velocity is bounded by 0.1m/s and 4 N is the maximum frictional force that could be applied without the twisting motion. Therefore, it fails to track a greater tension force.



Fig. 7: Snapshots of the multi-wire harnessing task, where the robot needs to harness two wires in a sequence. The robot first captures the workspace point cloud to identify and grasp the wire. Then, the proposed harnessing motion is applied to align the wire to the fixtures. Task 1 and 2 represent two different desired wire shapes and are executed sequentially.

The linear dynamics benchmark, on the other hand, struggles with stable tension tracking due to its inability to accurately capture the wire’s complex tension dynamics, underscoring the superiority of our proposed method.

Further investigation into harnessing velocity effects shows that varying maximum robot speeds yield comparable tension tracking performances, with a velocity of $0.1m/s$ facilitating the quickest task completion as shown in Fig. 6(b). Hence, this velocity limit was adopted for all subsequent wire-harnessing tasks.

B. Single-Wire Harnessing Task

Next, we evaluated our complete proposed pipeline on the wire harnessing task. As illustrated in Fig. 7, the robot first identifies the task board’s pose using an April Tag. Subsequently, the motion planning module designates a series of waypoints for the robot to follow. To infer the grasping pose, the robot captures a multi-view point cloud of the wire near the initial fixed point using its in-hand camera and then selects the highest point as the grasp location. This straightforward yet effective heuristic ensures reliable wire grasping, applicable even in scenarios involving multiple wires. After grasping, the robot proceeds along the planned waypoints, maintaining wire tension via the Koopman-based MPC, updating the fix-point as necessary, and employing insertion primitives to anchor the wire within the clamps.

	T1 Succ.Rate	Failures	T2 Succ.Rate	Failures
Proposed	10/10	N/A	10/10	N/A
No-Twist	0/10	B(10)	0/10	B(10)
Linear	5/10	A(2)B(2)D(1)	7/10	B(2)C(1)

TABLE I: **Single wire-harnessing success rates:** Our proposed approach and baselines were tested across two harnessing tasks in 5 trials each. Failure modes include: A (wire tangled with clamps), B (failed to insert the wire into clamps), C (wire was pulled out of the connector), and D (wire slipped from the gripper due to excessive force).

We designed two harnessing tasks that navigate the wire

from the same starting point to either a left or right route, depicted in Fig.1. Table. I presents the overall performance. Remarkably, our proposed method successfully completed all 10 trials for both tasks, significantly outperforming the baselines. The No-twisting approach, as detailed in Section V-A, could not generate sufficient tension using stretching alone, failing to robustly insert the wire into the clamps. Likewise, the Linear model struggled with stable force tracking, occasionally applying excessive tension or completely losing tension, making the wire difficult to insert into the clamps or slip out of the gripper. These outcomes further affirm the Koopman operator’s utility in capturing the intricate dynamics of wire tension.

C. Multi Wire Harnessing Task

	T1 Succ.Rate	Failures	T2 Succ.Rate	Failures
Proposed	9/10	A(1)	8/9	A(1)
No-Twist	0/10	B(10)	N/A	N/A
Linear	8/10	B(1)D(1)	6/8	A(1)D(1)

TABLE II: **Multi wire-harnessing success rates:** The robot harnesses two wires to solve task 1 and task 2 in a sequence. The robot executes task 2 only if the task 1 is successfully solved.

Lastly, we evaluated our method in a challenging multi-wire harnessing scenario, wherein the robot had to harness two wires into desired shapes for tasks 1 and 2. The snapshots of the harnessing sequence are shown in Fig. 7. As Table II illustrates, our method replicated the success of the single-wire task in the first harnessing attempt. However, the second task’s complexity increased due to the presence of an already inserted wire, which raises the likelihood of entanglement. Despite this, our method successfully completed 8 out of 9 trials for the second task, showcasing substantial robustness. In contrast, the No-twist baseline could not advance past the first task, and the Linear model baseline, struggling with tension regulation, achieved only 6 successful multi-wire harnessing out of 10 attempts.

VI. CONCLUSION

In this paper, we introduced a robust single-robot wire-harnessing pipeline, diverging from previous approaches that relied on dual-robot arms or tactile sensing. Our method employs the robot's twisting motion to generate the tension necessary for inserting the wire into clamps, utilizing only the built-in F/T sensor. The approach comprises four key components: 1) a Koopman operator-based MPC for wire following while maintaining the desired tension, 2) a harnessing motion planner to determine a sequence of waypoints, 3) a set of insertion primitives for securing the wire in clamps, and 4) a fix-point switching module to update the wire's constraints. We assessed our method using an industrial-level wire harnessing testboard, where it demonstrated robust performance and high success rates, and outperforms other baselines. The capability to handle multi-wire harnessing tasks further underscores the effectiveness of our method.

However, our method is not without limitations: 1) It assumes the wire is pre-inserted and an initial wire pose that is distanced from the clamps to avoid tangling with the task board. 2) The learned wire dynamics model is learned on a specific wire type and may not generalize directly to wires of different sizes and materials. 3) The continuous twisting and stretching motions could lead to material fatigue and damage. Future work will explore learning initial sorting and insertion motions through reinforcement or imitation learning to set the stage for the harnessing task. Additionally, we aim to develop online adaptation techniques to update the wire dynamics model in real-time [10], [25]–[27], enabling the system to adjust to new wires during operation. Also, the desired tension force can be adjusted during execution to minimize the potential wire damage, similar to [28].

REFERENCES

- [1] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka, "Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9881–9887.
- [2] X. Zhang, S. Jain, B. Huang, M. Tomizuka, and D. Romeres, "Learning generalizable pivoting skills," *arXiv preprint arXiv:2305.02554*, 2023.
- [3] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *ICRA 2022 Workshop: Reinforcement Learning for Contact-Rich Manipulation*, 2022. [Online]. Available: <https://openreview.net/forum?id=Zrp4wpa9lqh>
- [4] K. Galassi and G. Palli, "Robotic wires manipulation for switchgear cabling and wiring harness manufacturing," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2021, pp. 531–536.
- [5] F. Süberkrüb, R. Laezza, and Y. Karayiannidis, "Feel the tension: Manipulation of deformable linear objects in environments with fixtures using force information," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 216–11 222.
- [6] K. Chen, Z. Bing, Y. Wu, F. Wu, L. Zhang, S. Haddadin, and A. Knoll, "Real-time contact state estimation in shape control of deformable linear objects under small environmental constraints," *arXiv preprint arXiv:2401.17154*, 2024.
- [7] A. Wilson, H. Jiang, W. Lian, and W. Yuan, "Cable routing and assembly using tactile-driven motion primitives," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 408–10 414.
- [8] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, "Robotic cable routing with spatial representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5687–5694, 2022.
- [9] M. Yu, B. Liang, X. Zhang, X. Zhu, X. Li, and M. Tomizuka, "In-hand following of deformable linear objects using dexterous fingers with tactile sensing," *arXiv preprint arXiv:2403.12676*, 2024.
- [10] C. Wang, Y. Zhang, X. Zhang, Z. Wu, X. Zhu, S. Jin, T. Tang, and M. Tomizuka, "Offline-online learning of deformation model for cable manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5544–5551, 2022.
- [11] M. Yu, K. Lv, C. Wang, Y. Jiang, M. Tomizuka, and X. Li, "Generalizable whole-body global manipulation of deformable linear objects by dual-arm robot in 3-d constrained environments," *arXiv preprint arXiv:2310.09899*, 2023.
- [12] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1–8.
- [13] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1927–1934.
- [14] Assembly performance metrics and test methods. [Online]. Available: <https://www.nist.gov/el/intelligent-systems-division-73500/robotic-grasping-and-manipulation-assembly>
- [15] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3426–3433, 2018.
- [16] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, vol. 41, no. 6, pp. 599–614, 2022.
- [17] S. Jin, C. Wang, X. Zhu, T. Tang, and M. Tomizuka, "Real-time state estimation of deformable objects with dynamical simulation," in *Workshop on Robotic Manipulation of Deformable Objects*, 2020.
- [18] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 2372–2379, 2020.
- [19] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2146–2153.
- [20] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.
- [21] A. Mauroy, Y. Suzuki, and I. Mezić, *Koopman operator in systems and control*. Springer, 2020.
- [22] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the koopman operator and model predictive control," *arXiv preprint arXiv:1902.02827*, 2019.
- [23] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS one*, vol. 11, no. 2, p. e0150171, 2016.
- [24] Y. Li, H. He, J. Wu, D. Katabi, and A. Torralba, "Learning compositional koopman operators for model-based control," *arXiv preprint arXiv:1910.08264*, 2019.
- [25] X. Zhang, C. Wang, L. Sun, Z. Wu, X. Zhu, and M. Tomizuka, "Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning," in *7th Annual Conference on Robot Learning*, 2023.
- [26] Y. Sun, W. L. Ubellacker, W.-L. Ma, X. Zhang, C. Wang, N. V. Csomay-Shanklin, M. Tomizuka, K. Sreenath, and A. D. Ames, "Online learning of unknown dynamics for model-based controllers in legged locomotion," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8442–8449, 2021.
- [27] C. Wang, X. Zhang, Z. Kuang, and M. Tomizuka, "Safe online gain optimization for cartesian space variable impedance control," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 751–757.
- [28] X. Zhang, M. Tomizuka, and H. Li, "Bridging the sim-to-real gap with dynamic compliance tuning for industrial insertion," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4356–4363.