

# Rendering, Replicating, and Adapting Human Motions on a Cable Robot for Artistic Painting

Gerry Chen

*Institute for Robotics and Intelligent Machines (IRIM)  
School of Interactive Computing, College of Computing*

*Georgia Institute of Technology  
Atlanta, GA*

Thesis Committee:

Dr. Seth Hutchinson                          Dr. Jun Ueda  
IRIM, Interactive Computing                 IRIM, Mechanical Engineering  
*Georgia Institute of Technology*              *Georgia Institute of Technology*

Dr. Frank Dellaert                           Dr. Jean Oh  
IRIM, Interactive Computing                 Robotics Institute  
*Georgia Institute of Technology*              *Carnegie Mellon University*

Dr. Danfei Xu  
IRIM, Interactive Computing  
*Georgia Institute of Technology*

## Summary

In this proposed work, three parts to the problem of artistic painting guided by human motions are addressed: *rendering* a digital artwork in paint with a cable robot; *replicating* human input motions as closely as possible; and *adapting* human input motions to accommodate for differences in level of detail, style, and artistic medium according to high-level artist intents. Through the difficult, interaction-rich task of robot art, modern challenges in human-robot collaboration can be studied. In particular, techniques for robot motion control through natural input interfaces drawn from human motions are developed. Rendering paint requires advances in state estimation and control techniques for fast, fluid motions on a cable robot. Replicating human motions bridges the input motions and robot kino-dynamic capabilities, requiring advances in optimal trajectory re-timing techniques. Finally, adapting goes beyond rote replication by augmenting input motions to better fit the composition, style, and medium intended by the robot-artist team, requiring novel art-specific composition, text- and style- conditioned trajectory generation, and brush modeling techniques. Put together, the proposed thesis forms a cohesive body of work producing human-robot paintings and making novel contributions to the fields of robot art, human-robot collaboration, and cable robot control.

## CONTENTS

<b>1</b>	<b>Introduction</b>	1
I	Problem Relevance: Robot Painting	1
II	Overview of Approach	2
<b>2</b>	<b>Rendering</b>	3
III	Related Works	3
IV	Robot Hardware Design	3
V	Control and Estimation	4
VI	Summary	6
VII	Limitations	6
<b>3</b>	<b>Replicating</b>	6
VIII	Related Works	6
IX	Capturing Human Motions	7
X	Trajectory Retiming	7
XI	Summary	10
<b>4</b>	<b>Adapting</b>	11
XII	Related Works	11
XIII	Composition: Infills and Stroke Ordering	12
XIV	Generative Models for Painting Trajectories	12
XV	Generalization to Brush Painting Medium	14
XVI	Summary	14
<b>5</b>	<b>Sample Applications</b>	15
XVII	Public Painting on the Library	15
XVIII	Non-destructive Plant Phenotyping	15
<b>6</b>	<b>Conclusions</b>	15
<b>References</b>		16
<b>Appendix</b>		21
Appendix A: Cable Robot Additional Related Works		21
Appendix B: Cable Robot Control and Estimation Details		22
Appendix C: Expressing Cable Robot Constraints in the Task Space		26
Appendix D: Factor Graph LP Solving Details		26

# Chapter 1: Introduction

In striving for ever-deeper human connection, artists have pushed their crafts to embody the furthest reaches of human capabilities. From delicate painting motions to athletic ballet or parkour displays to dexterous musical performances, the diverse skills in art exhibit the widest reaches of physical ability and the most famed artists represent perhaps the most skilled humans in history. In addition to the pure artistic value of creating embodied machines that can perform at the highest levels of human artistry, it stands to reason that the study of robot art must also bring us closer to enabling robots that can do any physical task a human could.

To this end, we focus our work on the task of artistic *painting* guided by *human motions*. As further justified in I, we mainly focus on graffiti spray painting, but also study brush-based painting and calligraphy.

The choice to ground our robot in human motions is a both artistic and practical one. From an artistic point of view, we believe that motion is a much more intuitive, rich, and expressive form of communication for visual art than *e.g.* text, code, or combining artwork. Although we borrow ideas and techniques from generative AI for motion adaptation, we focus on the more raw communication modality of motion. From a practical point of view, human painting motions are excellent priors for creating human-like paintings and reproducing motions that better reflect the goal of creating robots whose physical abilities are strict supersets of human abilities.

At its core, art is the deepest form of communication, so studying how humans and robots can work together to create art is a natural way to study how to create more intuitive ways for humans to communicate with robots.

## I. PROBLEM RELEVANCE: ROBOT PAINTING

Artistic painting presents an excellent robotics task to study human motion adaptation because: (1) it has social implications, (2) it is easy to recognize execution errors but difficult to perform, and (3) there is a strong body of prior works and a high ceiling.

### A. Social Implications

As a form of communication, art has the potential to shape the way society perceives robots and the way robots become adopted in society. On a personal note, I hope to share my view that we should not fear robots as job stealers or existential threats, but instead embrace them as the super-powering tools they are, or perhaps even as “our children” as Marvin Minsky might argue.

Although graffiti is often pictured as an act of vandalism, it is in need of cultural preservation as an important art form that reflects the voices of disenfranchised populations. Much of the academic literature has investigated graffiti through the lens of youth, sub-culture, and the

discourse of the disenfranchised [1]. There has been enormous shift in the perception of the art form and, today, graffiti is not only increasingly appreciated by the society, but there are government funded graffiti projects and legislative efforts protecting graffiti arts [2]. The inherent ephemeral nature of graffiti makes graffiti particularly in need of preservation, and the statistically shorter life expectancy of those disenfranchised populations active in the graffiti community only exasperates the need for preservation efforts. As Michelangelo’s David is eternalized in Stone and da Vinci’s Mona Lisa is eternalized on poplar panel, we hope that someday the works of Keith Haring, Fabian “Occasional Superstar” Williams, and countless other named and unnamed artists will be immortalized in graffiti recreations.

Heroic efforts to eternalize graffiti by means such as photographing, capturing via 3D reconstruction, and motion recording have made great contributions to preserving this valuable art-form, but they do not aim to recreate graffiti artworks in their native formats: in paint on buildings, bridges, and other publicly accessible architectural structures. Paintings on such public displays present a core identity of graffiti in giving a voice to those who cannot otherwise be heard. John Morse captures the importance of public display well:

“People read these bandit signs. And they’ll read ‘em if they’re about an electrician; they’ll read ‘em if they’re about anything. If they read it and they like it, great. If they read it and they don’t like it, great. But the fact is: they’ll read it. They’re gonna read your poetry and that’s my goal: I want people to read my poetry.”

### B. Evaluation and Difficulty

Our brains are hardwired to interpret visual artwork, and we are able to recognize execution errors in paintings at a glance. Similar to how faces and hands can be very challenging for generative AI models, painting in general is a difficult task for robots not only due to the inherent technical difficulties, but also due to the high threshold for human acceptance. From an execution perspective (but not an artistic merit perspective), the ease of qualitatively evaluating whether a finished artwork matches the input intents, combined with our intuitive understanding for the problems when we see painted lines, make art a good task to advance robotics. As an example, the lack of stiffness of cable robots causes oscillations that are almost imperceptible to the human eye, but which are clearly revealed in paint as the spray paint can oscillates in reaction to disturbances. In a more personal anecdote, I laugh at my own attempts to draw human bodies since, after stepping back, it is so painfully obvious that the proportions are comically wrong.

Simultaneously, since many art forms capture the pinnacle of human physical capabilities – such as parkour’s athleticism, ballet’s precision, and calligraphy’s dexterity – creating art is a popular benchmark for robot skills because it is so difficult. Painting requires fine motor control and dexterity. Spray painting also requires highly dynamic motion control since the paint can must be moved at a precise speed to be dispersed on the canvas evenly, with human artists regularly reaching  $6\text{m/s}$  and  $50\text{m/s}^2$  [3]. Meanwhile, calligraphy and brush painting also require modeling of the complex, infinite-dimensional brush bristles and a controller that can compute the trajectory of the brush handle required to produce a desired image on the canvas. Finally, the embodiment of the robot artist painting in the real world presents all the usual practical challenges with building robust robots while also reinforcing social implications.

### C. Existing Research Area

Embodied artistic painting is an active research area in robotics. Brush-based painting in mediums such as acrylic [4]–[6], ink [7], oil, and water-color [8] have sought to create ever-more faithful depictions of reference photos. Similarly, brush-based calligraphy [9], [10] aims to devise accurate brush models and optimizers to produce long, beautiful brush strokes, especially strokes that have accurate beginning and ending shapes.

Artistic spray painting robots are also heavily studied, with many different platforms from manipulators [11] to gantry-based systems [8], [12] to mobile manipulators [13] to aerial robots [14]–[17] to cable robots [18]–[20]. A considerable amount of research is also dedicated to spray paint dispersion and motion planning for industrial applications (*e.g.* automotive painting and coating).

Purely software painterly rendering has also been long studied, dating back to image-gradient-based sketchification of images [21] to modern style transfer [22] and generative diffusion models [23] that can be zero-shot prompted with any style or medium imaginable. A vast set of techniques have now been investigated including greedy stroke-based rendering [24], deep reinforcement learning [25], pixel-based generative models [23], and many more.

The large activity in the space is driven by the great potential for improvement. Algorithmically generated SBRs are still readily distinguishable from a human painting due to the numerous small strokes and calligraphy brush models are currently insufficiently accurate especially without closed loop feedback. Meanwhile, spray painted art has nowhere close to the physical or artistic skills of human artists in terms of stroke variation and composition. Finally, generating robot painting *trajectories* is an important component of embodied AI painting that is overlooked by pixel-based approaches

such as neural style transfer (NST) and image diffusion models. The gap between human artists and robot artists is so great that we can safely plan for many more years of research in the area of robot painting, and the skills required to create a robot artist will almost certainly teach us about creating robots that can operate in kitchens, healthcare, and other domains.

## II. OVERVIEW OF APPROACH

We organize our presentation of results in tackling robot painting into 3 parts:

**Rendering** (the “hand”) – First, we describe our custom robot platform and the novel advancements required to execute the highly dynamic motions required of graffiti spray painting. We describe hardware design, state estimation, and control of our cable-driven parallel robot (CDPR). The design requirements of graffiti art informed our decision to use a CDPR design, which can economically scale to large sizes and achieve high speeds, but at the costs of requiring more advanced controllers and lacking stiffness. Our novel factor graph -based state estimator and controller combined with our high-bandwidth communication link produces superior tracking performance enabling successful execution of spray painting trajectories.

**Replicating** (the “spinal cord”) - Next, we discuss how human motions can be captured and replayed on the CDPR. We use both motion capture and tablet-based input devices, but still require preprocessing steps before feeding trajectories to the CDPR tracking controller, including path smoothing and trajectory re-timing to retarget human motions to the cable robot to minimize tracking errors associated with low stiffness.

**Adapting** (the “brain”) - Finally, we investigate the ways human inputs may be modified to produce high-quality artwork from low-detail inputs, such as text prompts or rough sketches. We study artistic composition as path planning problems to allow sparse drawings to be filled-in as complete artworks; generative modeling to generate robot painting trajectories directly from high-level inputs; and adaptation to the brush painting medium to show how our generative approaches can apply to other dynamical systems.

The layout of this proposal is as follows. Chapters 2–4 go through these 3 parts, respectively, in detail and with their own related works sections. Chapter 5 will discuss applications of our research including public painting demonstrations and a hydroponic plant phenotyping project sharing many of the same technologies. Finally, Chapter 6 will conclude with a summary of our contributions and future directions.

## Chapter 2: Rendering

To render graffiti spray painting, we choose to use a CDPR because they scale well to large sizes, achieve fast speeds and accelerations, and are relatively low-cost. However, cable robots also present challenges associated with high-bandwidth control and lack of stiffness.

In this section, we will discuss selection criteria (including related works), robot design, and novel techniques for control and estimation.

### III. RELATED WORKS

#### A. Robot Platforms for Spray Painting

Several robots have been previously been developed for spray painting. Serial arm manipulators and gantry-based systems are precise and mature, but arms do not scale well to large workspaces [11], [26] and gantry-based systems exhibit a tradeoff between size and portability [8], [12]. Mobile manipulators address these issues, but are currently not as dynamic or precise as human artists [13]. Aerial robots are popular for their ability to paint otherwise inaccessible walls, but have been cited as being difficult to accurately control due to susceptibility to disturbances and comparatively limited acceleration capabilities [14]–[17]. Cable-based systems appear to be promising, but so far [18], [19] have only demonstrated raster- or stippling- style painting while [20] has not demonstrated the highly dynamic motions employed by human artists. The use of only 2-cables limits the dynamic capabilities of the cable robot since cable suspended parallel robots (CSPRs) are under-constrained, *i.e.* the set of wrenches that can be applied to the end-effector does not contain an arbitrarily small hyper-sphere centered at the origin, however 4-cable robots are sufficiently constrained for planar motion and are thus able to achieve higher speeds and accelerations.

Therefore, we choose to use a 4-cable CDPR design for our graffiti spray painting robot.

#### B. CDPR State Estimation

CDPR state estimation is a prerequisite for control and has been thoroughly studied in the contexts of pose estimation, offline calibration, and online/continuous calibration and state estimation. Pose estimation, provided good calibration parameters, can in most cases be computed by solving a nonlinear least squares (NLS) problem with Gauss-Newton iteration [27]. Offline self-calibration can be done by performing a pre-defined movement routine [28], [29] or by passively collecting sensor data and performing an optimization to compute the parameters [30], [31]. Online continuous calibration continuously updates calibration parameters but can be challenging when the data does not permit well-conditioned optimization problems requiring special considerations. For more detail, see Appendix A.

#### C. CDPR Tracking Control

CDPR control has evolved from basic inverse kinematics in the 1980s to modern methods like dual-space PID, feedback linearization, SMC, and MPC. While linear methods like PID are common, they can be sensitive to robot configuration [32]. SMC is favored for robustness but commonly has “chattering” issues [33]. MPC provides strong performance but poses implementation challenges [34]. For additional detail, see Appendix A.

### IV. ROBOT HARDWARE DESIGN

Given the design requirements for painting graffiti based on human spray painting data, we believe that a CDPR is an ideal platform. In this section we detail our robot hardware.

#### A. Design Considerations

The primary design requirements for a graffiti painting system involve workspace size, maximum end-effector velocity, and maximum end-effector acceleration. We seek a platform which can be scaled to a workspace 20mx20m or larger, though in this work we only seek a demonstration sized at a few meters. Based on analysis from motion capture recordings of graffiti artists, we determined that we require 6m/s and 50m/s<sup>2</sup> of speed and acceleration, respectively. Assuming the mass of the spray can and actuating accessories do not exceed 2kg, including gravity the robot should be capable of exerting 120N upward and comparable forces in other directions.

Secondary design requirements include portability, accuracy, and stiffness. It should be feasible to disassemble and reassemble the robot on-site at the wall of a building. Accuracy and stiffness are considered secondary constraints because, compared to art forms such as brush painting or sculpture, graffiti is less sensitive to positional inaccuracies and experiences less reaction force. Based on the thickness of a line painted with a “needle” nozzle 5cm from the painting surface, we estimate 2.5cm of repeatability to be sufficient. We estimate an accuracy of 1% the size of the painting to be sufficient, based loosely on [35]. We estimate external disturbances to be negligible based on paint reaction forces and historical Atlanta wind speeds.

CDPRs present ideal platforms for graffiti painting given the aforementioned requirements. A CDPR is a robot whose end-effector is pulled by a set of cables which are driven by winches on a fixed base. Due to properties of cables, CDPRs can scale to extraordinary sizes and speeds [36], [37], albeit with reduced stiffness. These qualities make them ideally suited to the large but relatively undisturbed setting and modest accuracy requirement of graffiti painting.

CDPRs also have an active research community which has solved many challenges in workspace analysis [38]–

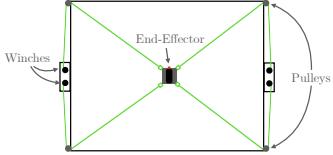


Fig. 1. Our planar CDPR has a 4-cable, rectangular configuration with the end-effector in the center carrying the spray paint can.



Fig. 2. Our cable robot (left) includes an end effector that carries the spray paint and actuator electronics (center) and 4 winch assemblies, each consisting of a shared motor controller, motor, and helical winch (right, x2).

[40], control, and estimation (further discussed in Sections III-B and III-C). Preliminarily, based on [41], we estimate a 1kHz update frequency to be necessary for real-time control.

Finally, we define the requirements to actuate the spray can nozzle. For a full can of Montana BLACK 400mL, the force required to depress the nozzle was measured to be 27N and the displacement was measured to be 2mm. Other 400mL spray cans by the brands Montana, Hardcore, and Kobra were found to have similar actuation forces and displacements.

### B. Approach

Our CDPR uses 4 cables in a planar configuration to exert pulling forces on the end effector via 4 motor-driven winches (see Fig. 1). The end effector was built to be lightweight and carry the spray can and actuating electronics. It has 4 mounting points to connect to the 4 CDPR cables. The cables are low-stretch and lightweight (cable sag is negligible). The spray can nozzle actuating mechanism is wireless, battery-powered, and im-

plemented using a servo with the lever-arm mechanism from [42]. Complete design details can be found in our accompanying arXiv paper [43].

### C. Results

Our assembled robot is pictured in Figures 3 and 2. The winches satisfy our design requirements with each being capable of pulling a 2kg mass on the cable up to 7.6m/s and 94m/s<sup>2</sup> and bidirectionally communicating at 1kHz. The end-effector and spray can actuating mechanism are also pictured in Fig. 2. The total mass varied between 1006g and 1317g depending on the spray can. The spray can actuating mechanism was able to depress the spray nozzle with 100% success rate in a trial of 100, 1 second long actuations. The latency from commanding to dispensing paint was measured to be 400ms.

### D. Discussion & Limitations

We are able to paint well, as in Fig. 3, despite not being able to use our 6DoF captured data to its full potential since we are limited to planar motion. Also, we will discuss in Section X that the paint limits us to a maximum speed far below what the hardware is capable of (and the speed suggested by the motion capture analysis). A combination of hardware upgrades and intelligent paint modeling and optimization are likely necessary to leverage our system's full potential, especially actuation to move the nozzle *closer to the canvas*.

## V. CONTROL AND ESTIMATION

We present a novel CDPR state estimator and tracking controller utilizing factor graphs which performs an offline nonlinear optimization whose solution is used to generate an online, optimal, linear state estimator and controller. The state estimator uses only the current measurements and future works can use the same framework to extend the state estimator to also perform online continuous calibration. Because the factor graph offline models probabilistic uncertainties, the linear online continuous calibration would only require current measurements while also naturally balancing old and new data. Meanwhile, the linear controller is generated from a nonlinear optimization offline which makes it optimal, while being easy to implement online by virtue of the fact that it is a linear, time-varying PD controller.

We frame our approach around factor graphs because we believe them to be useful abstractions for describing trajectory optimization and controls topics. Simultaneously, factor graphs are highly efficient, visually insightful, and modular, allowing efficient incorporation of nearly any sensor, control objective, and/or constraint with minimal modification [45]. Probabilistic graphical models, including factor graphs, have seen success in state estimation and perception [46] and are becoming increasingly popular as tools for optimal control [44],



Fig. 3. Photo of a graffiti painting by our robot.

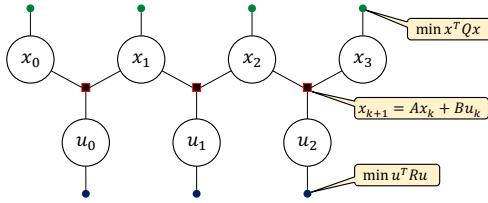


Fig. 4. Prototypical example of a factor graph describing the LQR problem with 4 timesteps, adapted with permission from [44]. Large circles are variables and small black circles/squares are factors.  $x_k$  represent state variables,  $u_k$  represent control variables, black squares represent dynamics constraint factors, and black circles represent optimal control objectives.

[45], [47]. Factor graphs, in short, are a graphical way to describe optimization problems where variables (unknowns to solve for) and factors (optimization objectives or constraints) are connected by an edge if the variable is involved in the objective/constraint. Fig. 4 shows a prototypical example of a factor graph describing the LQR problem.

Our algorithm, summarized in Fig. 5, consists of three offline stages followed by an online controller.

#### A. Trajectory Generation and Gain Matrix Pre-Computation

Offline, we can compute the optimal nominal trajectory then, almost for free, we get a (time-varying) locally-optimal Kalman Filter and LQR controller. The algorithm is summarized:

- Pre-compute a **nominal trajectory** that tracks the reference trajectory by “solving” a factor graph. This is equivalent to trajectory generation using the iterative Linear Quadratic Regulator (iLQR) algorithm [48].
- Pre-compute **time-varying LQR gain matrices** by linearizing the graph around the nominal trajectory and using *variable elimination* on the linearized graph. Linearizing the graph is equivalent to applying a first-order Taylor expansion to convert the constrained *nonlinear* least squares problem into a constrained *linear* least squares problem (finite horizon LQR problem), and variable elimination is equivalent to using backward-induction to solve the Bellman equation.
- Pre-compute **time-varying Kalman Filter (KF) gain matrices** by creating a graph with stochastic factors instead of cost/constraint factors, *linearizing* the graph around the nominal trajectory, and marginalizing at each time step. The stochastic factors represent noise in the dynamics and/or measurements, and marginalizing at each time step is the Markov assumption the KF makes.

Full details can be found in Appendix B or in [49].

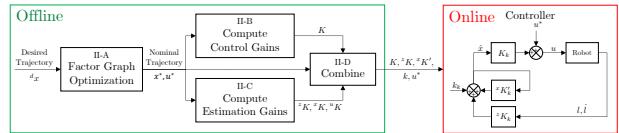


Fig. 5. Our proposed controller leverages offline computations to simplify the online controller, which is locally optimal while requiring only 3 vector-matrix multiplications and 3 vector additions per update. The offline stage, outlined in green, generates time-varying gain matrices and feedforward vectors. The online stage, outlined in red, is a standard TV-LQG controller.

The result is, for each timestep  $k$ , a nominal point  $x_k^*$ ,  $u_k^*$  and gain matrices  $K_k$  for the LQR controller and  $z K_k$ ,  $x K_k$ ,  $k K_k$  for the Kalman Estimator.

#### B. Online LQG Controller

Finally, we combine the LQR and KF gain matrices to form an online controller in the form of a TV-LQG controller. This is of course thanks to the standard LQG result: the stochastic optimal control policy is to feed the optimal estimator into the optimal controller, for systems with only state- and control- independent noise. From the Kalman Filter, we obtain an optimal estimate  $\hat{x}_k$  for  $\delta x_k$  so LQG states that we may use  $\hat{x}_k$  in place of  $\delta_x$  from the LQR controller:

$$\delta u_k = K_k \hat{x}_k \quad (1)$$

$$\hat{x}_k = {}^x K'_k \hat{x}_{k-1} + {}^u K_k \delta u_{k-1} + {}^z K_k \delta z_k \quad (2)$$

where  $\delta z_k$  denotes the deviation in cable lengths and speeds from the nominal values computed by IK on  $x^*$ .

Making the assumption that our controller always follows the optimal controller, we can also substitute (1) into (2):

$$\hat{x}_k = {}^x K'_k \hat{x}_{k-1} + {}^z K_k \delta z_k \quad (3)$$

where  ${}^x K'_k := {}^x K_k + {}^u K_k K_k$ .

And finally, for completeness, we can substitute our  $\delta$  expressions to obtain our final time varying LQG controller:

$$\hat{x}_k = {}^x K'_k \hat{x}_{k-1} + {}^z K_k z_k + k_k \quad (4)$$

$$u_k = K_k \hat{x}_k + u_k^* \quad (5)$$

where  $k_k := - {}^z K_k z_k^*$ .

The only computations that need to be run online are (4) and (5), which require only 3 matrix multiplications and 3 vector additions per update. Meanwhile,  $\mathcal{K}_k := K_k, {}^x K_k, {}^z K_k, {}^u K_k, u^*, k_k$  are pre-computed offline.

*1) Interpolation:* It may be desirable to run the controller at a faster rate than the set of precomputed  $\mathcal{K}$ , since  $\mathcal{K}$  may be only slowly changing while we typically want a controller update rate of at least several hundred Hz. We can devise a simple “zero-th” order interpolation:

$$\hat{x}(k dt + \Delta t) = {}^x K'_k \hat{x}_{k-1} + {}^z K_k z(k dt + \Delta t) + k_k \quad (6)$$

Although we found in our experiments that this interpolation is sufficient, more intelligent interpolation schemes may be investigated.

## VI. SUMMARY

In this section, we presented our robot platform which can execute given trajectories at high speeds and accelerations and scale to large sizes. We described the graffiti spray painting criteria that informed our choice of CDPR platform and design parameters. We also presented our novel CDPR state estimator and tracking controller algorithm which leverages factor graphs to produce optimal estimators and controllers which are easy to implement online. Although the CDPR state estimator currently estimates only the end effector pose but not the kinematic or dynamic parameters, it could be extended to also estimate the kinematic and dynamic parameters online. Finally, although the CDPR is planar, it could be naturally extended to SE(3) by adding another symmetric set of 4 cables, or at minimum 3 additional cables in a non-symmetric pattern.

## VII. LIMITATIONS

Although our planar CDPR can successfully track desired trajectories, it still suffers from relatively low stiffness which can cause (out-of-plane) oscillations from disturbances which are exacerbated by excessive accelerations. In particular, minor imbalances out-of-plane in the end effector center of mass, which are unavoidable as the paint can empties over time changing the mass distribution, cause out-of-plane disturbances in reaction to in-plane accelerations. Although certain out-of-plane modes are in theory controllable (but not reachable) through internal cable tension and input shaping, in practice they are virtually impossible to model, observe, and control. Instead of rigorously modeling the out-of-plane dynamics, for which even producing high-quality input disturbances with actuator bandwidth limitations is challenging [50], [51], we instead make the simple heuristic that minimizing in-plane accelerations will also minimize out-of-plane oscillations.

Additionally, paint deposition is sensitive to the trajectory speed profile. A maximum speed constraint is imposed to deposit a sufficient thickness of paint.

In the next section, we will discuss how we capture and modify human input trajectories so that they are *replicated* as faithfully as possible on the cable robot while still satisfying the aforementioned speed and acceleration constraints of the robot and paint deposition.

## Chapter 3: Replicating

We wish to replicate a human artist’s vision for an art piece as faithfully as possible. As mentioned in Chapter 1, we choose to guide our robot with *human motions*, since they are intuitive, rich, and expressive while also serving as excellent priors for more general robotic tasks.

We formulate our replication problem as follows: given a human input motion, CDPR limits, and paint deposition characteristics, we wish to record and follow the input motion path exactly using the robot and with the best feasible paint deposition speed that respects all limits. The process of replicating human motions is then naturally composed of 2 parts: (1) capturing motions and (2) modifying their timings to accommodate CDPR and paint deposition dynamics.

Capturing human motions is relatively straightforward: we study using both a motion capture system and an iPad as human motion input devices, and describe the procedures for each in Section IX.

Captured motions must be modified because cable robots inherently lack stiffness and paint deposition has speed constraints, so naïvely applying captured motions to the robot will result in shaking and uneven paint deposition. Modifying captured motions to satisfy constraints while still being as faithful as possible to the original human input is framed as a trajectory retiming problem, subject to speed, acceleration, and torque constraints/objectives.

Specifically, we formulate the Time-Optimal Path Parameterization (TOPP) problem in task space for our cable robot, implement the TOPP-RA algorithm, then extend the algorithm to *Quadratic Objective Path Parameterization* (QOPP) using factor graphs to balance multiple the multiple objectives of painting speed, control margins, and end effector stiffness. We describe our work in Section X.

## VIII. RELATED WORKS

The trajectory retiming problem, in which we seek a dynamically feasible speed profile for a given path, is a well-studied problem in robotics. In addition to applications such as ours in which the task pre-determines the path we must take and we need only compute a timing, it is also useful in the decoupled approach to trajectory generation. The decoupled approach is commonly studied in high-dimensional systems (*e.g.* humanoids) because it computes first a kinematically feasible path, then a dynamically feasible timing, making it more tractable than the full kino-dynamic trajectory optimization problem [52, Ch. 11.2].

The *time-optimal* trajectory retiming problem (or Time-Optimal Path Parameterization, TOPP) represents perhaps the most common retiming objective and was practically solved in the late 1980s, although

algorithmic enhancements have continued to be made. The time-optimal objective is to minimize the time to traverse the path. It has been studied extensively in the literature and traditionally has taken one of three approaches [53]: convex optimization, dynamic programming, and searching for bang-bang control switching points where the active constraints change.

Two key precursor techniques enabling efficient algorithms are shared by almost all approaches, dating back to at least [54]–[58]. First, the equations of motion and constraints are reparameterized in terms of the *scalar* time parameterization function (see Sec. X-B1 for details). This enables Bellman-style forward-backward algorithms. Second, the monotonicity of the time-optimal objective implies that the solution must lie on the boundary of the feasible set (bang-bang). Hauser [59], Nagy & Vajk [53], and Pham et. al. [53] all utilize proofs along these lines to justify the use of sequential linear programming (SLP) or greedy speed maximization. This bang-bang approach is efficient but restricts the objective to minimum-time or similar objectives.

However, the *time-optimal* retiming problem is not always the most desirable objective [60], particularly in applications where we seek to balance multiple objectives or where bang-bang control is unsuitable. For example, cable-driven parallel robots maintain stiffness primarily through internal tension which diminishes the closer they are to torque limits. Thus balancing speed with torque margin is desirable to maintain stability and safety [3], [49]. This and several other applications in balancing robot safety, stability, and wear with speed of operation motivate the use of quadratic objectives, which can minimize the sum of multiple squared errors instead of or in addition to hard constraints with TOPP.

Surprisingly alternate retiming objectives are rarely considered in the literature. Dynamic programming approaches [56], [58], [61] discretize not only in time but also in state space. Some approaches address other objectives, especially energy-minimization [62], [63], but apply only to specific objectives, *e.g.* integral of a time-independent running cost. Direct transcription approaches tend to be the most general [60], [64] but, even with second-order cone problem or sparse linear algebra solvers, do not fully exploit the structure of this scalar-function optimal control problem.

## IX. CAPTURING HUMAN MOTIONS

The capture process is important for both learning about the motions to produce graffiti art and as an input modality for artists to create new art. As such, we use both motion capture, which is highly accurate but expensive and cumbersome, and a tablet, which is less accurate but more accessible and portable. Although we do not quite bring the project to this point, in principle

the former modality provides enough information for us to later augment iPad and low-effort motions with the sufficient spatial and stylistic information to produce high-quality graffiti art from low-effort input data.

In this work, we consider 3 different capture strategies:

- 1) collect a library of composable shape outlines,
- 2) record the motions for a full painting, and
- 3) capture motions to be painted in real-time.

For (1), we seek to record the full SE(3) trajectory of the spray can as an artist is actively painting, so we use motion capture for its high accuracy, precision, framerate, and ease of use. We rigidly attach motion capture markers to a spray paint can and canvas to record graffiti artist collaborators’ SE(3) painting trajectories. The 2 artists were each instructed to paint the outlines of the 26 letters of the alphabet and a handful of shapes of their choosing. The trajectory data was cleaned and processed to obtain a library of shape outlines. Further details on data collection procedure and results can be found in [3, Sec II].

For (2) and (3), we seek a more accessible, lower-cost capture method so we opt to use a tablet (iPad) to record 2D artist motions. Drawings are recorded in the browser, making it portable to any tablet or computer, but at the expense of losing some data due to security restrictions (*i.e.* timestamp accuracy is limited to 1ms or worse on most browsers due to Spectre and Meltdown). As a result, some minor pre-processing is needed to smooth trajectories.

## X. TRAJECTORY RETIMING

We wish to faithfully reproduce the human’s input motion while also respecting the acceleration and speed constraints described in Section VII. To remain faithful to the original human motion, we seek to only *retime* the trajectory while retaining the original path shape. To accommodate the low stiffness inherent to cable robots, we seek to impose both task-space acceleration and joint-space torque limits / objectives. To accommodate the paint deposition constraints, we seek to impose task-space speed limits / objectives. Finally, to minimize painting time for both drip minimization and economy, we seek to impose a trajectory duration objective.

From these considerations, we introduce Quadratic Objective Path Parameterization (QOPP): an extension of Time-Optimal Path Parameterization (TOPP) to include quadratic objectives thereby allowing balancing multiple objectives. Figure 6 shows how retiming improves the cable robot’s ability to replicate the input path.

Our three contributions are as follows:

- 1) *Applying TOPP-RA to our Graffiti Robot:* Our first contribution is a simple demonstration that TOPP-RA can be effectively applied to the graffiti robot domain. Given input trajectories, we first parameterize each



Fig. 6. Cable robot executing the same path for the letter “G” using the human-input timestamps (left) and using our retimed timestamps (right) shows that retiming the trajectory according to speed and acceleration limits avoids the inaccuracies and oscillations that occur when excess speeds and accelerations are commanded by the naïve human operator. Images were generated by overlaying video frames to simulate a long-exposure image.

stroke as a spline using either regression or interpolation on a subset of waypoints. We then apply TOPP-RA to each spline to obtain a time-optimal parameterization of the trajectory. Finally, we execute the trajectory on the robot using the controller described in Section V.

2) *Relationship to Factor Graph Elimination:* Our second contribution is in demonstrating an equivalence between the state-of-the-art TOPP-RA algorithm and factor graph elimination. In particular, we can show that the TOPP-RA algorithm is equivalent to eliminating a factor graph representing the well-known projected path constraints (see [65, Eq. 5, 6]), with the  $t = T, \dots, 0$  elimination ordering. The proof is very similar to our published work in [45], except we must extend our factor graph elimination to include inequality linear constraints to solve the TOPP problem. In the general case this is Fourier-Motzkin Elimination, whose complexity can grow exponentially w.r.t. the number of elimination steps [66]. However, it can be shown that in the special case when inequality constraints appearing in the “joint” after eliminating each variable only ever apply to one scalar variable, the number of inequality constraints never grows. The TOPP problem has this exact special structure which is leveraged by both the factor graph elimination algorithm and by TOPP-RA, and in fact it can be shown that the two algorithms are completely equivalent.

3) *Quadratic Objective Path Parameterization (QOPP):* Our final contribution is to extend TOPP problem approaches to quadratic objectives to enable balancing multiple objectives. The factor graph interpretation allows us to naturally extend to arbitrary objectives and constraints, as long as we can define an elimination operation on those factors. We derive and implement a method to perform exact elimination for the retiming problem with quadratic objectives, linear inequalities, and linear equalities. We apply the algorithm for our cable robot painting problem and show that balancing objectives generates more accurate paintings than TOPP.

### A. TOPP Formulation for Graffiti

We define our TOPP problem drawing from the notation and work of [65] as follows. Given a path in task space  $\mathbf{T}(s) : [0, 1] \rightarrow SE(3)$  with twist  $\dot{\mathbf{x}}(s) \in \mathbb{R}^6$  and corresponding  $\mathbf{q}(s) : [0, 1] \rightarrow \mathbb{R}^n$  in joint space, we seek to find a re-parameterization,  $s(t)$  is a monotonically increasing function from  $[0, T] \rightarrow [0, 1]$ , which is time-optimal, i.e. one which minimizes the total time  $T$  while satisfying a set of constraints:

$$s^*(t) = \arg \min_{s(t)} T \quad (7a)$$

subject to

$$\mathbf{A}(s)\ddot{\mathbf{x}} + \dot{\mathbf{x}}^T \mathbf{B}(s)\dot{\mathbf{x}} + \mathbf{f}(s) \in \mathcal{C}(s), \quad (7b)$$

$$\mathbf{A}^v(s)\ddot{\mathbf{x}} + \mathbf{f}^v(s) \in \mathcal{C}^v(s) \quad (7c)$$

where all the constraints are applied for all  $t \in [0, T]$ ;  $\mathbf{q}$ ,  $\dot{\mathbf{x}}$ , and their derivatives are functions of  $s(t)$  although arguments are omitted for readability;  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{f}$  denote coefficient matrices/tensors/vectors for general first and second order constraints; and  $\mathcal{C}$  denote convex polytopes of admissible values for the corresponding constraints.

The constraints (7b) and (7c) can represent all the constraints we seek to impose, namely task-space speed and acceleration limits (natively) and joint-space acceleration limits through a straightforward transformation. To express cable robot joint constraints in the task space, we start with the equations of motion for a free-floating body (the end-effector) and substitute the winch dynamics and motor torque constraints. Additional details are provided in Appendix C.

### B. TOPP Solution using Factor Graphs

We solve the TOPP problem by applying the same method outlined in [65]. This relies on 3 key insights:

- 1) The task-space constraints in (12) can be re-written in terms of  $s(t)$  and its derivatives.
- 2) The clever parameterizations  $x = \dot{s}^2$  and  $u = \ddot{s}$  allows us to rewrite the  $s(t)$  constraints as a scalar linear dynamical system with linear control-action constraints.
- 3) The linear dynamical system forms a linear program which can be solved in linear time using a dynamic programming algorithm, which is equivalent to variable elimination.

We now describe each of these steps.

- 1) *Re-writing Constraints in Terms of  $s(t)$ :* Differentiating  $\mathbf{x}(s(t))$  with respect to  $t$  yields

$$\dot{\mathbf{x}}(s(t)) = \frac{d\mathbf{x}}{ds} \dot{s} \quad (8)$$

$$\ddot{\mathbf{x}}(s(t)) = \frac{d^2\mathbf{x}}{ds^2} \dot{s}^2 + \frac{d\mathbf{x}}{ds} \ddot{s}. \quad (9)$$

Substituting into (7b) and (7c) yields constraints of the form

$$\mathbf{a}(s)\ddot{s} + \mathbf{b}(s)\dot{s}^2 + \mathbf{c}(s) \in \mathcal{C}(s) \quad (10)$$

$$\mathbf{a}^v(s)\dot{s} + \mathbf{c}^v(s) \in \mathcal{C}^v(s) \quad (11)$$

where  $\mathbf{a}(s), \mathbf{b}(s), \mathbf{c}(s), \mathbf{a}^v(s), \mathbf{c}^v(s)$  are vector functions of  $s$ .

Because  $\dot{s}$  is a scalar function, (11) can be rewritten in the form of (10). Specifically, (11) can be interpreted as intersections of a translated ray with a convex region, which in turn can be reduced to just a convex set in  $\mathbb{R}^1$ . Furthermore, because  $s(t)$  is monotonically increasing, we have another constraint that  $\dot{s}(t) > 0$  so we can safely square the constraint. For example, if (11) reduces to  $\dot{s} \in (a, b)$ , then we can write  $0\ddot{s} + 1\dot{s}^2 + 0 \in (\min(0, a)^2, \min(0, b)^2)$ . Therefore, we have turned our TOPP problem into:

$$s^*(t) = \arg \min_{s(t)} T \quad (12a)$$

$$\text{subject to } \mathbf{a}(s)\ddot{s} + \mathbf{b}(s)\dot{s}^2 + \mathbf{c}(s) \in \mathcal{C}(s) \quad (12b)$$

2) *Parameterization:* At this point, to remove the dependence of the coefficients on  $s$ , we discretize our problem in  $s$ :

$$a_i \ddot{s}_i + b_i \dot{s}_i^2 + c_i \in \mathcal{C}_i \quad (13)$$

$$T = \int_0^1 \frac{1}{\dot{s}(t)} ds \approx \sum \frac{2\Delta_s}{\dot{s}_i + \dot{s}_{k+1}} \quad (14)$$

where the summation holds exactly for the piecewise-constant assumption on  $\dot{s}$  [67, Sec 6.1.1].

Defining  $x := \dot{s}^2$  and  $u := \ddot{s}$ , we can rewrite (12b) as

$$\mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \in \mathcal{C}_i \quad (15)$$

and, remarkably,  $\frac{dx}{ds} = 2\dot{s}\frac{d\dot{s}}{ds} = 2\frac{d\dot{s}}{ds}\frac{ds}{dt} = 2\ddot{s} = u$ . Then assuming a zero-order hold on  $u$ , we have the relationship between  $x$  and  $u$ :

$$x_{i+1} = x_i + 2u_i \Delta_s. \quad (16)$$

Finally, it is proved in [65] that the objective  $\min T$  can be achieved by greedily selecting the maximum  $\dot{s}_i$  from the set of admissible values at each time step. Following the same reasoning, it is possible to prove that the TOPP problem is equivalent to the following LP:

$$\underset{\substack{x_0, \dots, x_N, \\ u_0, \dots, u_{N-1}}}{\text{maximize}} \quad \sum_{i=0}^N w^i x_i \quad (17a)$$

subject to

$$\mathbf{a}_i u_i + \mathbf{b}_i x_i + \mathbf{c}_i \in \mathcal{C}_i, \quad i = 0, \dots, N, \quad (17b)$$

$$x_{i+1} - x_i - 2u_i \Delta_s = 0, \quad i = 0, \dots, N-1, \quad (17c)$$

$$x_i > 0, \quad i = 0, \dots, N. \quad (17d)$$

for some very large  $w$ , which denotes that each  $x_i$  should be taken greedily and irrespective of any other  $x_i$ .

Intuitively, the equivalence of the optimization problems despite a different objective function is due to the fact that there exists some critical threshold  $\Delta_s^*$  such that, when  $\Delta_s \leq \Delta_s^*$ , the dynamics coefficient  $2\Delta_s$  will be smaller than  $a_i/b_i$  for all  $i$  so that it will never be better to sacrifice a smaller  $x_i$  to gain a larger  $u_i$  (which in turn gains a larger  $x_{i+1}$ ). The details of the proof require additional machinery to address the fact that  $a_i/b_i$  is not defined for vectors, but the intuition is the same.

3) *Solving the LP:* Although [65] solves the LP in  $\mathcal{O}(N)$  time using a reachability analysis approach, we can instead use factor graph elimination to derive an equivalent algorithm.

Solving the LP (17) is a straightforward extension of the same approach used in [3], [44], [45] and summarized in Figs. 7–9. First, the LP (17) is expressed as a factor graph (Fig. 7). Next,  $u_0$  is eliminated by substituting the “dynamics” equation (17c). Then,  $x_0$  is eliminated by greedily taking the largest feasible value for  $x_0^*$  and solving 2, 2-variable scalar LPs to determine the minimum and maximum bounds on  $x_1$  given the constraints on  $x_0$ , obtaining Fig. 8. Eliminating the remaining variables proceeds the same way, finally resulting in the Bayes Net (Fig. 9), from which the solution  $x^*, u^*$  is obtained via back-substitution, involving 1, 1-variable scalar LP solve per timestep. Full details can be found in Appendix D or [68].

Finally, the optimal time parameterization  $s^*(t)$  can be obtained by integrating  $x^* = \dot{s}^*$ . We defer to [69] for the intricacies of parameterizing solution. As in [69], zero-inertia points are accurate in the limit  $\Delta_s \rightarrow 0$ . The time optimal trajectory is  $\mathbf{q}^*(t) = \mathbf{q}(s^*(t))$ .

### C. Extending to Quadratic Objectives

The variable elimination algorithm naturally extends to other objectives because it remains unchanged no matter the objectives or constraints; only the algebra of each elimination step changes. Let us then define our (discretized) general quadratic objective problem as:

$$\underset{\substack{x_0, \dots, x_N, \\ u_0, \dots, u_{N-1}}}{\arg \min} \quad \sum_{i=0}^N \tilde{\mathbf{x}}_k Q_k \tilde{\mathbf{x}}_k + \tilde{\mathbf{u}}_k R_k \tilde{\mathbf{u}}_k + \tilde{\mathbf{x}}_k N_k \tilde{\mathbf{u}}_k \quad (20a)$$

$$\text{subject to } (17b), (17c), (17d) \quad (20b)$$

where scalars  $\tilde{\mathbf{x}}_k := x_k - x_{k,\text{desired}}$ ,  $\tilde{\mathbf{u}}_k := u_k - u_{k,\text{desired}}$ , and  $Q_k, R_k, N_k$  are state, control, and cross cost weights.

To apply variable elimination to this new problem, we must define how to eliminate variables when they may have quadratic objectives, linear equalities, and/or

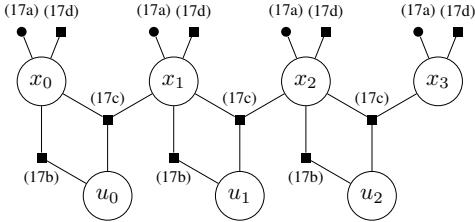


Fig. 7. Factor graph for a 4-timestep TOPP problem. Each variable node represents a variable  $x_i$  or  $u_i$  in the LP. Each factor node represents a constraint (square) or objective (dot) term.

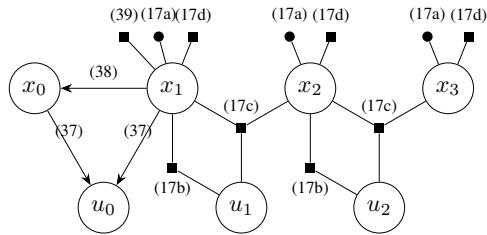
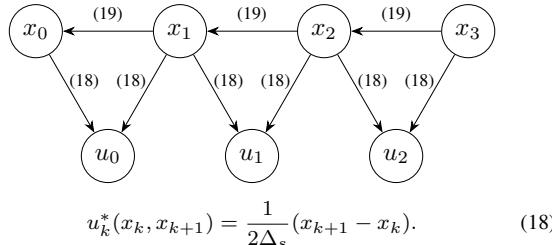


Fig. 8. Factor graph for a 4-timestep TOPP problem after eliminating  $u_0, x_0$ . The arrows denote back-substitution to be done at the end and the new factor (39) denotes the constraint propagated to  $x_1$  after eliminating  $x_0$ .



$$u_k^*(x_k, x_{k+1}) = \frac{1}{2\Delta_s} (x_{k+1} - x_k). \quad (18)$$

$$\left. \begin{aligned} x_k^*(x_{k+1}) &= \max_{x_k} x_k \\ \text{s.t. } & \frac{\mathbf{a}_k}{2\Delta_s} (x_{k+1} - x_k) + \mathbf{b}_k x_k + \mathbf{c}_k \in \mathcal{C}_k, \\ & x_{k,\min} \leq x_k \leq x_{k,\max}, \\ & x_k > 0. \end{aligned} \right\} \quad (19)$$

Fig. 9. After eliminating all the variables, we obtain a *Bayes Net*. Arrows denote conditionals (18)(19) which we can efficiently back-substitute.

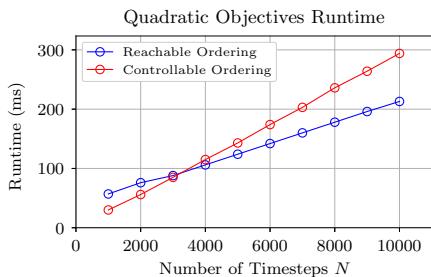


Fig. 10. Runtime plot for a sample quadratic objective retiming problem shows that our algorithm is still  $\mathcal{O}(n)$ , even with quadratic objectives.

linear inequalities. Our problem is much easier than in the general case because the retiming problem involves only scalars.

Eliminating any variable with an equality factor (*e.g.*  $u_k$ ) is easy and remains largely unchanged from the TOPP case. The only difference is that, when creating the new joint factor on the separator, we substitute into not only the inequalities but also into the quadratic objectives to form a new quadratic objective.

Eliminating a variable without an equality factor, *i.e.* one with quadratic objectives and linear inequalities, is a parametric QP problem. The solution  $x_k^*(S(x_k))$  turns out to piecewise linear, making the joint piecewise quadratic. Furthermore, it is straightforward to compute in the scalar case by finding the parametric unconstrained solution to the quadratic objective (the solution is linear) then computing the intersections with the inequality constraints and traversing the inequality constraints to the minimum and maximum bounds of  $S(x_k)$ . The full details are available in our paper [68].

Although theoretically our QOPP algorithm's complexity is quadratic with the number of timesteps, in practice we find the number of segments each timestep usually remains bounded (around 20) so it maintains linear time as in TOPP (Fig. 10).

#### D. Other Possible Extensions

Although not addressed in this thesis, extending beyond quadratic objectives should be possible in a number of ways, including handling nonlinear objectives and constraints, obtaining optimal feedback gains, solving full kinodynamic motion planning by alternating path planning and retiming, and incrementally reparameterizing for real-time control.

#### XI. SUMMARY

In this part, we showed that (1) motions directly recorded with motion capture or tablet input device cannot be faithfully executed at the recorded speed due to cable robot dynamics limitations, (2) applying trajectory retiming using the time-optimal objective produces much better results, and (3) extending the time-optimal objective to a quadratic objective allows balancing multiple objectives producing even better results. Figure 6 well-illustrates the improvement in replication accuracy resulting from our trajectory retiming algorithm.

## Chapter 4: Adapting

In this part, we aim to address three objectives. First, we aim to provide utility to artists by extending beyond replicating their motions to also modifying/building on top of them, thereby cooperatively amplifying their creativity. Second, we explore a high-level strategy to generate robot trajectories that generalizes to different platforms and inputs. Finally, we explore the different facets of style and how we may disambiguate them from content.

Because art is so broad, we must restrict our scope to a few well-posed sub-problems that are both useful and feasible:

- 1) Composition: given shape outlines, generate the necessary motions and order of motions to fill-in and/or outline the shapes.
- 2) Restylization: given vague or rough artist intent, *e.g.* conveyed by a prompt, use generative models to produce robot painting trajectories.
- 3) Medium Transfer: given an artwork in one medium or abstraction, generalize artworks to the brushed paint medium by learning a brush model.

These three avenues build on each other to satisfy our objectives. Composition represents, in conjunction with existing works in paint modeling and stroke-based rendering, a way to generate training data. More specifically, it is a bridge between the large but unlabeled painting trajectory datasets and the vast body of existing research in pixel-based image understanding. Restylization applies generative models to painting trajectories including studies of how to disambiguate style from content and other dimensions of intent. Finally Medium Transfer demonstrates generalization of the approach to a new medium and robot platform.

## XII. RELATED WORKS

### A. Stroke-Based Rendering

Stroke-Based Rendering (SBR) is the task of replicating a reference image as closely as possible using a sequence of (brush) strokes. Aaron Hertzmann was an early pioneer in SBR dating back to at least the late 1990s [24]. His approach to greedily place brush strokes one at a time in the region with the greatest error from the reference image is still popular today, but modernized with differentiable renderers [70], RL [71], GANs [72], and more.

Because SBR requires a rendering model, many modern Stroke-Based Rendering works also train a small neural network to predict the shape of a brush stroke given some trajectory parameters. A common approach is to execute a number of painting stroke trajectories and, for each stroke, image the resulting stroke on the

page and use supervised learning to predict the rendered stroke from the trajectory parameters [5].

One limitation of SBR is that it tends to produce many short strokes instead of few long strokes. Perhaps this is because long strokes introduce too much correlation across long time and space horizons making the problem too challenging without good initializations as in the calligraphy application.

### B. Generative Sketch Models

A wide variety of more modern approaches have been studied for generating drawing trajectories, especially from inputs other than reference images.

Sketch-RNN [73] is a seminal and prototypical example of an autoregressive approach which uses a sequence-to-sequence VAE (using RNN encoders/decoder) to generate a sequence of strokes one at a time from a latent vector derived from an input sequence “prompt” trained on a dataset of human-drawn sketches [74]. Many works use Sketch-RNN’s VAE-RNN architecture [75], [76]. GANs are also popular, sometimes combined with VAEs [77], [78]

Non-autoregressive approaches generate all the strokes at once instead of one stroke at a time. CLIPDraw [79], StyleCLIPDraw [80], CLIPasso [81], and CLIPascene [82] are examples of optimization-based approaches where a differentiable renderer [83] enables optimizing for an input trajectory whose CLIP-embedded rendering best matches a CLIP-embedded language prompt [84]. Several recent works have expanded upon this by incorporating image diffusion models to help better initialize and guide the differentiable renderer [85]–[87], but interestingly do not directly apply diffusion to a vector/trajecotry representation.

Unlike classic SBR, generative models tend to produce a very different style of stroke-based rendering. Namely, they tend to produce free-hand sketches [88] instead of painterly renderings. Characterized by fewer longer strokes, these low-detail drawings have a high degree of abstraction. In some ways, these sketches are better suited to our objective of adapting human motions which often come in the form of similar sketches, but we also seek to impart more detail to the inputs so as to complement the artist rather than replace them.

### C. Generative Robot Trajectories

While a number of LLM works have shown LLMs effective at planning [89], [90], we instead focus on approaches that directly generate robot actions.

Auto-regressive approaches for generating robot trajectories appear to be the most common and, when conditioned on some input, can be interpreted as policies. Transformers are popular for their performance and ease of conditioning on various inputs. RT-1 [91] is a seminal

example which uses a decoder-only transformer to output a (single) tokenized action from a tokenized input sequence computed from vision and language inputs, an approach shared by many papers. RT-2 [92] instead fine-tunes an existing Large Vision-Language Model by tokenizing actions by existing language tokens (*e.g.* 1-1000 or 256 least-frequently-used tokens) creating a multi-modal vision-language-action model. Several works use LLMs to generate executable code representing policies or trajectories [93]–[95]. Various Sequence2Sequence approaches [73], [96], [97] are also used for generating open-loop trajectories. Several works also use an RL approach with a variety of architectures [98].

Non-autoregressive models include Diffusion, VAEs, and GANs. The seminal paper series [99], [100] uses diffusion with a temporal convolution backbone to generate robot trajectories conditioned via inference-time guidance or classifier-free guidance.

### XIII. COMPOSITION: INFILLS AND STROKE ORDERING

Through real-world testing, we found that artists using our graffiti painting robot system with the iPad input device typically only draw simple shape outlines but do not “color-in” or do shading. We hypothesize this to be because the unavoidable disconnect between the artist iPad inputs and the robot painting on the canvas limits the interaction to the high-level, abstract shapes instead of the low-level, detailed shading.

Therefore, we propose to automatically complete high-level shape outline inputs with algorithmically generated infills and shading. This will enable artists to focus on the high-level shapes and let the robot fill in the details. Because filling in the shapes may require long times for large murals, we propose allowing the artist to input all the shape outlines upfront and allow the system to reason about the order stroke trajectories must be executed.

This is implemented in [3], whereby the artist inputs shape outlines using motion capture (or from a pre-recorded library of captured shapes as described in Section IX). The scale and pose of each shape is placed by the artist on a virtual canvas in order of “depth”. The infill is then computed as an application of a classic path planning coverage algorithm [101], [102]. Finally, we use the simple stroke order of painting the infill of each shape before its outline, and painting the shapes from back to front, *i.e.* the shape furthest in the background is painted first, and the shape closest to the foreground is painted last.

Future works may consider algorithms such as hidden line removal or hidden surface determination [103] to reduce the paint which is wasted on occluded regions and reduce the number of paint color changes required by painting all of the same color at once.

### XIV. GENERATIVE MODELS FOR PAINTING TRAJECTORIES

Leveraging recent advancements in generative models for creating robot trajectories has been of great interest recently. With the successes of image diffusion models and large language models (LLMs) which can generate remarkably good results to remarkably diverse input scenarios, many roboticists are now excited about the potential of these approaches in enabling robots to operate with less structured information, more human-like intuition, and natural language interaction. Seminal diffusion policy papers [99], [100], [104] show diffusion models can apply to robot motions. Meanwhile, seminal embodied LLM papers [89], [90] show LLMs are adept at high-level planning [105], although direct action prediction using LLMs remains less mature (so we will not make use of LLMs for trajectory generation directly, but may use their multi-modal capabilities for image interpretation).

However, obtaining the requisite extremely large scale datasets needed to train these models is a challenge for many robotic tasks [93], [106]. For example, generative image models such as DALL-E 2 and StableDiffusion rely on the CLIP model (which was trained on a dataset of 400 million image-text pairs) to correspond text to images, and some experts believe that the state-of-the-art LLMs are trained on the entirety of the internet’s high-quality data. Even the recent robotic manipulation dataset “Open X-Embodiment” [106], a collaboration across dozens of academic institutions, contains on the order of only 1 million trajectories (compared to the 400 million of CLIP or estimated 13 trillion for GPT-4). Generating sufficient data to correspond robot trajectories to another modality is a major roadblock in applying generative models to create robot trajectories directly, without an intermediate policy for specific tasks.

We propose circumventing this roadblock for robotic art by leveraging infilling (Section XIII) and pre-existing SBR work as a *bridge* between robot trajectories and generative pixel-based art, which has large amounts of data and pre-trained models freely available for corresponding images to text and other modalities. We anticipate that the pair will enable novel results in generating robot trajectories from text and other high-level, abstract descriptions. The inspiration for this research stems from the intriguing intersection of visual information and motion that painting offers. By leveraging the extensive body of work in text-to-image synthesis, we aim to bridge the gap between textual descriptions and robot trajectories. This approach not only offers a novel perspective on robot motion generation but also taps into the vast potential of existing research in the text-to-image domain.

## A. Proposed Approach

We propose training a trajectory diffusion model conditioned on artist intents (text and/or sketches) using unlabeled painting trajectory data augmented by compositional techniques and rendered image language correlations. In this section, we will outline the proposed approach with preliminary justifications as well as alternative implementation choices to investigate.

*Architecture:* We intend to use trajectory diffusion with classifier-free guidance [107] as in Decision Diffuser [100]. Diffusion strikes a good balance between expressiveness and ease of use (as compared to GANs and VAEs), especially for the natural coarse-to-fine structure of visual art. The temporal convolution approach in [100] also allows for arbitrary-length trajectory generation. The choice of classifier-free guidance over guided diffusion is less obvious, but the difficulty of evaluating style on noisy samples [107] combined with the slightly improved performance demonstrated in action planning [100] suggests classifier-free diffusion may be superior. Inference-time guidance may still be considered, for example to satisfy auxiliary conditional objectives not trained on or to apply diffusion in the image space and constrain on trajectories simulated by a differentiable renderer from Section XV. As compared to auto-regressive approaches, diffusion models may be easier to train.

*Training Data / Generating a shared language embedding:* Training data in the form of artistic trajectories already exists in various forms but is (1) not labeled with the information we seek and (2) low quality. For example, the #000000book (“black book”) [108] contains  $> 70000$  trajectories of (unlabeled) graffiti tags and Google’s *Quick, Draw!* [74] contains 50 million doodles of 345 object classes, but neither contain stylistic annotations and both are composed of line drawings as opposed to full artworks. Therefore, our biggest challenges are 2-fold: (1) corresponding the trajectories with the semantic information we seek and (2) improving the quality of the data.

Two directions for generating a shared embedding between trajectories and language are:

1) **Stroke-Based Rendering (Text→Image→Path):**

Stroke-based rendering (SBR) enables generating painting strokes that produce a desired image. Applying SBR to the outputs of pre-existing image diffusion models could produce robot trajectories corresponding to image diffusion model inputs through the text→image→strokes pipeline. One major limitation of this approach is that SBR traditionally makes a series of small, point-like strokes rather than the large, brush-like strokes that are the more natural and characteristic of graffiti strokes we seek.

2) **Painting Simulation (Path→Image→Text):** Starting from an unlabeled dataset of painting strokes, we can render how the strokes would appear if painted then correspond the images with text via existing shared embeddings such as CLIP [84] or with a vision-language model such as GPT-4V [109]. Furthermore, a differentiable renderer could be combined with a diffusion-based planner in place of a traditional optimizer, combining the approaches in [83], [99]. This approach represents the opposite pipeline as the previous approach: strokes→image→text.

The Path→Image→Text approach appears more tractable given the difficulty of high-quality SBR. Although the opposite approach would likely enable more diverse data, higher quality textual understanding, and a potential alternate conditioning method (condition on an image), these may be partially offset by multimodal LLMs [109], [110] which can now enable labeling images with high-quality textual descriptions.

If time permits, the quality of the data may be improved by augmenting the trajectory data with the procedurally generated compositions from Section XIII. Alternatively, the Text→Image→Path approach may have the potential to generate higher quality data since the resulting simulated images emulate full artworks instead of line drawings. Instead of generating additional training data, HITL-TAMP [111] gives evidence that keeping the classical stroke planner from Section XIII while using a learned model for creating strokes may be a more efficient way to learn to paint. Finally, preference learning and representation learning have been shown to generalize trajectory quality ratings to new styles [112], which could be used to curate a dataset of good augmentations or SBR, even across different graffiti styles.

*Disambiguating Style and Content:* In addition to text prompts, we may also seek to learn style embeddings that can be extracted from images and conditioned on. Although style is difficult to quantitatively define, several proxies may be considered:

1) **Language:** Language can naturally describe both style and content (e.g. “Paint a cat in the style of Picasso”), so with a sufficiently strong textual embedding, our approach may be able to disambiguate style and content.

2) **Neural Style Transfer:** Neural style transfer (NST) [22], [113] is a popular technique for transferring the texture from one image to the content of another, but is most commonly understood for images. [114], [115] approximate the NST optimization with a trained neural network conditioned with a FiLM-like approach. Assuming texture to be a good approximation for style, NST with FiLM could

be applied to produce a conditioning space which could more easily be correlated with language than raw images or trajectories.

- 3) **Artist Embeddings:** If we have sufficient amounts of trajectory data labeled by artist identity, an artist embedding could be learned to discern the styles of different artists à la [116], and interpolating through the embedding space could produce trajectories in various styles.

Due to the lack of artist attribution in the datasets, the NST approach is favorable over the artist embedding approach. We also anticipate stylistic language descriptions to be learned with a sufficiently large (unlabeled) dataset. Nevertheless, we hope to study how other disentangling approaches may be used to isolate artist intents in the trajectory or image space.

## XV. GENERALIZATION TO BRUSH PAINTING MEDIUM

Thus far, our work has focused on graffiti spray painting whose rendering model is very simple: paint shows up as a rounded line with thickness. Through experimentation in Chapter 2, we found that for most nozzle shapes, high-quality artist spray paint brands (Montana, MTN, or Black), and the right speed, the paint dispersion pattern is sufficiently uniform that the paint can be assumed to make a solid circle around the center point of the nozzle whose diameter is directly proportional to the distance from the canvas. Furthermore, in Chapter 3, we have honed our cable robot sufficiently that any specified path can be accurately executed/replicated with an appropriate speed. Therefore, we have made the spray painting model no more challenging than traditional drawing models used in generative models for “doodles”/sketches/markers which use the same “line with thickness” model [73]. However, we are still interested in the more complex dynamics associated with robot embodiment and different mediums.

To show our approach can generalize to more other robot applications, we propose extending our approach to brush painting. Specifically, we focus on the calligraphy problem, to study how complex brush dynamics can be modeled and optimized over long time horizons without lifting the brush. Brush painting has a complex mapping from trajectory to image and an inverse map does not necessarily exist. Style is also heavily present in calligraphy, with many believing one’s writing is a reflection of one’s personality and temperament and with some per-calligrapher datasets available on Kaggle and Github. Incorporating both the brush and robot dynamics into the generative model will be challenging, but we believe the approach will be more representative of other robot applications.

Observations of artists suggest that medium transfer is both feasible and artistically utile. We have observed that many artists draw differently on an iPad than with spray point or with brushes. However, highly trained artists can be simultaneously proficient in multiple mediums: capable of expressing the same idea in multiple mediums and even emulating a variety of styles on a variety of mediums. Nevertheless, we posit that artists have preferred mediums and must learn to adapt their style to other mediums. The fact that artists have preferred mediums presents an *accessibility* motivation for designing a system to transfer artwork across mediums: opening up new mediums to artists can enable new avenues of expression, creativity, and collaboration. Simultaneously, the fact that artists have the ability to train to express the same compositions in different mediums suggests that the task of transferring artwork across mediums is feasible.

The data generation component of our aforementioned approach will be modified by extending the work of [9] by using a deep-learned brush model in place of the hand-crafted one. In contrast to previous works which take images of completed brush strokes, we will take videos of brush itself during painting to capture its shape dynamics. By collecting video data from multiple viewpoints of a brush executing various strokes, we propose training a recurrent neural network to predict the state (shape) of the brush given the previous state and relative motion. The state will be represented as a latent vector and the observed camera viewpoints will represent observed measurements. To then virtually render a painting for a given brush trajectory, we predict the contact patch of the brush with the canvas at each time instance and combine across time steps to produce a rendered image.

## XVI. SUMMARY

In conclusion, this chapter sets the stage for a ground-breaking exploration into the direct generation of robot trajectories for artistic painting motions. By leveraging generative models and building upon existing text-to-image research, we aim to redefine the boundaries of robot motion generation in the artistic domain.

## Chapter 5: Sample Applications

### XVII. PUBLIC PAINTING ON THE LIBRARY

This project, in collaboration with Georgia Tech Artist-In-Residence Tristan Al-Haddad, aims to create a large-scale mural on a 2-story, 50m-wide bank of windows on the north elevation of the Price Gilbert Library (Fig. 11 left) using our graffiti painting cable robot. Due to ventilation limitations, we replace the spray paint end-effector with a serial manipulator-mounted paint brush (Fig. 11 right).

The project showcases our cable robot's ability to paint large-scale artworks with mediums other than just spray paint. It attests to the ability of our replication and rendering pipeline to reproduce a variety of motions at a vastly magnified scale, and to the robustness and reliability of our robot painting platform.

### XVIII. NON-DESTRUCTIVE PLANT PHENOTYPING

This project, in collaboration with the Nutrients, Energy, and Water (NEW) Center for Agriculture Technology led by Professor Yongsheng Chen, applies our cable robot platform to hydroponic plant monitoring [117].

Using our cable robot, we enable high-accuracy, medium-throughput non-destructive plant phenotyping: a compromise between traditional high-throughput low-accuracy drones and low-throughput high-accuracy turntables. Because our painting robot scales well to large sizes, we can replace the spray paint can with a manipulator-mounted camera to monitor plants in large hydroponic systems. The cable robot enables covering a large workspace while the manipulator enables close-up inspection of individual plants from a variety of viewpoints. At our small-scale pilot site, we autonomously captured 150 images for each of 56 plants every day for 6 weeks without any human supervision or intervention. Using the images, we computed 3D reconstructions which were used to estimate plant mass as the plants grew to help scientists in the NEW Center develop better plant growth models.

The system evidences how developing a large-scale robot platform for painting has incidental applications in unexpected domains. It also demonstrates the robustness of our CDPR control and estimation algorithms, which effortlessly adapt to the high-stiffness, low-speed, long-term operation of plant phenotyping.

Both sample applications employ a serial manipulator on the low-stiffness cable robot end effector, for which future work could employ a closed-loop controller to stabilize the brush tip / camera similar to [118] and/or input shaping of manipulator movements to minimize nominal cable robot oscillations.

## Chapter 6: Conclusions

I have proposed studying the task of robot graffiti painting in 3 parts: rendering, replicating, and adapting. These tasks are a natural progression towards the overall objective of creating an embodied AI artist that can receive high-level artist inputs and produce graffiti art with paint. Furthermore, each of the parts require solving novel robotics challenges with applications beyond graffiti painting.

Through my proposed thesis, I will contribute to the body of knowledge in multiple fields. In addition to the field of robot art, I will make technical contributions to cable robot control, motion planning, and generative modeling. Finally, through collaborations, I also contribute to plant phenotyping research and public art.

Robot Graffiti exemplifies artistic expression and robot ambition. Between creating a new, accessible avenue through which artists may make public statements and pushing the boundaries of robot capabilities, the proposed thesis forms a cohesive, important, and well-scooped contribution to the field of robotics.



Fig. 11. Left: The bank of windows at Price Gilbert Library we will paint a mural on. Right: The cable robot positions a wooden carriage on which a serial manipulator with brush is mounted.

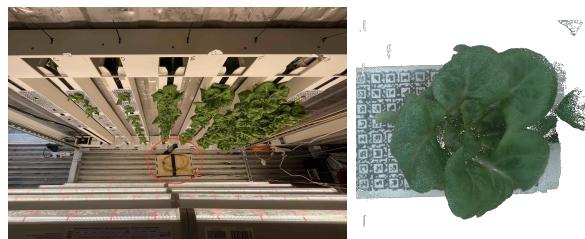


Fig. 12. Left: Our cable robot system with manipulator-mounted camera monitoring plants in a hydroponic system. Right: A sample point cloud of a plant.

## REFERENCES

- [1] U. Frederick, “Revolution is the new black: Graffiti/art and mark-making practices,” *Archaeologies*, vol. 5, pp. 210–237, 2009.
- [2] L. MacDowall, “In praise of 70k: Cultural heritage and graffiti style,” *Continuum*, vol. 20, no. 4, pp. 471–484, 2006.
- [3] G. Chen, S. Baek, J.-D. Florez, W. Qian, S.-W. Leigh, S. Hutchinson, and F. Dellaert, “GTGraffiti: Spray painting graffiti art from human painting motions with a cable driven parallel robot,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4065–4072. [Online]. Available: [https://gerry-chen.com/publications/Chen22icra\\_GTGraffiti/Chen22icra\\_GTGraffiti.pdf](https://gerry-chen.com/publications/Chen22icra_GTGraffiti/Chen22icra_GTGraffiti.pdf)
- [4] P. Schaldenbrand and J. Oh, “Content masked loss: Human-like brush stroke planning in a reinforcement learning painting agent,” 2021.
- [5] P. Schaldenbrand, J. McCann, and J. Oh, “Frida: A collaborative robot painter with a differentiable, real2sim2real planning environment,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 712–11 718.
- [6] A. Karimov, E. Kopets, S. Leonov, L. Scalera, and D. Butusov, “A robot for artistic painting in authentic colors,” *Journal of Intelligent & Robotic Systems*, vol. 107, no. 3, p. 34, 2023. [Online]. Available: <https://doi.org/10.1007/s10846-023-01831-4>
- [7] A. Bidgoli, M. L. De Guevara, C. Hsiung, J. Oh, and E. Kang, “Artistic style in robotic painting: a machine learning approach to learning brushstroke from human artists,” in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 412–418.
- [8] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, “Water-colour robotic painting: a novel automatic system for artistic rendering,” *Journal of Intelligent and Robotic Systems*, pp. 1–16, 2018.
- [9] S. Wang, J. Chen, X. Deng, S. Hutchinson, and F. Dellaert, “Robot calligraphy using pseudospectral optimal control in conjunction with a novel dynamic brush model,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6696–6703.
- [10] S. Mueller, N. Huebel, M. Waibel, and R. D’Andrea, “Robotic calligraphy - learning how to write single strokes of Chinese and Japanese characters,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 1734–1739.
- [11] L. Scalera, E. Mazzon, P. Gallina, and A. Gasparetto, “Airbrush robotic painting system: Experimental validation of a colour spray model,” in *Advances in Service and Industrial Robotics*, C. Ferraresi and G. Quaglia, Eds. Cham: Springer International Publishing, 2018, pp. 549–556.
- [12] N. Roy, “Graffiti robot - train writing - planet256 / Niklas Roy / the fly - wall printer style machine,” <https://www.youtube.com/watch?v=zSIdvQsu27s>.
- [13] Y. Jun, G. Jang, B.-K. Cho, J. Trubatch, I. Kim, S.-D. Seo, and P. Y. Oh, “A humanoid doing an artistic work - graffiti on the wall,” *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1538–1543, 2016.
- [14] A. Uryasheva, M. Kulbeda, N. Rodichenko, and D. Tsetserukou, “DroneGraffiti: Autonomous multi-UAV spray painting,” in *ACM SIGGRAPH 2019 Studio*, ser. SIGGRAPH ’19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: <https://doi.org/10.1145/3306306.3328000>
- [15] A. S. Vempati, M. Kamel, N. Stilinovic, Q. Zhang, D. Reusser, I. Sa, J. Nieto, R. Siegwart, and P. Beardsley, “PaintCopter: An autonomous UAV for spray painting on three-dimensional surfaces,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2862–2869, 2018.
- [16] B. Galea, E. Kia, N. Aird, and P. G. Kry, “Stippling with aerial robots,” in *Computational Aesthetics (Expressive 2016)*, 2016, p. 10 pages.
- [17] TsaruRobotics, “Autonomous mural for sprite Ukraine: <https://tsuru.su/en/project/spritemural/>. [Online]. Available: <https://tsuru.su/en/project/spritemural/>
- [18] “Albert robot muralist,” <https://www.robotmuralist.com/albert>. [Online]. Available: <https://www.robotmuralist.com/albert>
- [19] A. Liekens, Kenny, and L. Scheire, “Zet Kenny binnenkort zélf zijn eerste graffiti op een muur? — Team Scheire #9 [Will Kenny soon put his first graffiti on a wall? — Team Scheire #9],” <https://www.youtube.com/watch?v=jOpgvW7aIhQ>, Sept 2020.
- [20] J. Lehni, “Soft monsters,” *Perspecta*, vol. 40, pp. 22–27, 2008. [Online]. Available: <http://www.jstor.org/stable/40482274>
- [21] M. P. Salisbury, M. T. Wong, J. F. Hughes, and D. H. Salesin, “Oriental textures for image-based pen-and-ink illustration,” in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’97. USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 401–406. [Online]. Available: <https://doi.org/10.1145/258734.258890>
- [22] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *CoRR*, vol. abs/1508.06576, 2015. [Online]. Available: <http://arxiv.org/abs/1508.06576>
- [23] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2021.
- [24] A. Hertzmann, “Painterly rendering with curved brush strokes of multiple sizes,” in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998, pp. 453–460.
- [25] J. F. J. Mellor, E. Park, Y. Ganin, I. Babuschkin, T. Kulkarni, D. Rosenbaum, A. Ballard, T. Weber, O. Vinyals, and S. M. A. Eslami, “Unsupervised doodling and painting with improved spiral,” 2019.
- [26] D. Berio, S. Calinon, and F. F. Leymarie, “Learning dynamic graffiti strokes with a compliant robot,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3981–3986.
- [27] A. Pott, *Cable-Driven Parallel Robots: Theory and Application*, 1st ed. Springer Publishing Company, Incorporated, 2018.
- [28] P. H. Borgstrom, B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, “NIMS-PL: A cable-driven robot with self-calibration capabilities,” *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1005–1015, 2009.
- [29] W. Kraus, V. Schmidt, P. Rajendra, and A. Pott, “System identification and cable force control for a cable-driven parallel robot with industrial servo drives,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5921–5926.
- [30] X. Jin, J. Jung, S. Y. Ko, E. Choi, J.-O. Park, and C.-S. Kim, “Geometric parameter calibration for a cable-driven parallel robot based on a single one-dimensional laser distance sensor measurement and experimental modeling,” *Sensors*, vol. 18, no. 7, p. 2392, 2018.
- [31] P. Miermeister, A. Pott, and A. Verl, “Auto-calibration method for overconstrained cable-driven parallel robots,” in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.
- [32] J. Lamaury and M. Gouttefarde, “Control of a large redundantly actuated cable-suspended parallel robot,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4659–4664.
- [33] J.-J. E. SLOTINE, “Sliding controller design for nonlinear systems,” *International Journal of Control*, vol. 40, no. 2, pp. 421–434, 1984. [Online]. Available: <https://doi.org/10.1080/00207178408933284>
- [34] J. Cavalcanti Santos, “Model predictive tracking control of cable-driven parallel robots : From concept to real-time validation,” Ph.D. dissertation, University of Montpellier, 2020. [Online]. Available: <http://www.theses.fr/2020MONT017/document>
- [35] L. Della Valle, T. G. Andrews, and S. Ross, “Perceptual thresholds of curvilinearity and angularity as functions of line length.” *J Exp Psychol*, vol. 51, no. 5, pp. 343–347, May 1956.
- [36] R. Nan, D. Li, C. Jin, Q. Wang, L. Zhu, W. Zhu, H. Zhang, Y. Yue, and L. Qian, “The five-hundred-meter aperture spher-

- ical radio telescope (FAST) project,” *International Journal of Modern Physics D*, vol. 20, no. 06, pp. 989–1024, 2011.
- [37] S. Bandyopadhyay, “Lunar crater radio telescope (LCRT) on the far-side of the moon,” [https://www.nasa.gov/directories/spacetech/niac/2020\\_Phase\\_I\\_Phase\\_II/lunar\\_crater\\_radio\\_telescope/](https://www.nasa.gov/directories/spacetech/niac/2020_Phase_I_Phase_II/lunar_crater_radio_telescope/), April 2020.
- [38] P. Bosscher, A. T. Riechel, and I. Ebert-Uphoff, “Wrench-feasible workspace generation for cable-driven robots,” *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 890–902, 2006.
- [39] S. Bouchard, C. Gosselin, and B. Moore, “On the ability of a cable-driven robot to generate a prescribed set of wrenches,” *Journal of Mechanisms and Robotics*, vol. 2, no. 1, 2/15/2021 2009. [Online]. Available: <https://doi.org/10.1115/1.4000558>
- [40] M. Gouttefarde, D. Daney, and J. Merlet, “Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots,” *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 1–13, 2011.
- [41] M. Gouttefarde, J. Lamaury, C. Reichert, and T. Bruckmann, “A versatile tension distribution algorithm for  $n$ -dof parallel robots driven by  $n+2$  cables,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1444–1457, 2015.
- [42] A. Liekens, “Spray can servo mount,” <https://www.thingiverse.com/thing:4622176>, Oct 2020.
- [43] G. Chen, S. Baek, J.-D. Florez, W. Qian, S. won Leigh, S. Hutchinson, and F. Dellaert, “Extended version of GTGraffiti: Spray painting graffiti art from human painting motions with a cable driven parallel robot,” 2021, arXiv:2109.06238 [cs.RO]. [Online]. Available: <https://arxiv.org/pdf/2109.06238.pdf>
- [44] G. Chen and Y. Zhang, “LQR control using factor graphs,” <https://gtsm.org/2019/11/07/lqr-control.html>, Nov 2019, note: Superceding ICRA 2021 paper pending review.
- [45] S. Yang, G. Chen, Y. Zhang, H. Choset, and F. Dellaert, “Equality constrained linear optimal control with factor graphs,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9717–9723. [Online]. Available: [https://gerry-chen.com/publications/Yang21pre\\_ecLQR/Yang20pre\\_equalityconstrainedLQR.pdf](https://gerry-chen.com/publications/Yang21pre_ecLQR/Yang20pre_equalityconstrainedLQR.pdf)
- [46] F. Dellaert and M. Kaess, “Factor graphs for robot perception,” *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.
- [47] F. Dellaert, “Factor graphs: Exploiting structure in robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 141–166, 2021. [Online]. Available: <https://doi.org/10.1146/annurev-control-061520-010504>
- [48] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems,” in *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO*, INSTICC. SciTePress, 2004, pp. 222–229.
- [49] G. Chen, S. Hutchinson, and F. Dellaert, “Locally optimal estimation and control of cable driven parallel robots using time varying linear quadratic gaussian control,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7367–7374. [Online]. Available: [https://gerry-chen.com/publications/Chen22iros\\_cdpr-control/Chen22iros\\_cdpr\\_tracking\\_control.pdf](https://gerry-chen.com/publications/Chen22iros_cdpr-control/Chen22iros_cdpr_tracking_control.pdf)
- [50] Y. Qiu, M. Wu, L. H. Ting, and J. Ueda, “Maximum spectral flatness control of a manipulandum for human motor system identification,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3271–3278, 2021.
- [51] Y. Qiu, M. Wu, L. H. Ting, and J. Ueda, “Optimal multisine perturbations for improved dynamic system identification using a mechanical platform: A preliminary simulation study,” in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2023, pp. 948–953.
- [52] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005.
- [53] Á. Nagy and I. Vajk, “Sequential Time-Optimal Path-Tracking Algorithm for Robots,” *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1253–1259, Oct. 2019.
- [54] J. Bobrow, S. Dubowsky, and J. Gibson, “Time-Optimal Control of Robotic Manipulators Along Specified Paths,” *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, Sep. 1985. [Online]. Available: <https://doi.org/10.1177/027836498500400301>
- [55] K. Shin and N. McKay, “Minimum-time control of robotic manipulators with geometric path constraints,” *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, Jun. 1985.
- [56] F. Pfeiffer and R. Johanni, “A concept for manipulator trajectory planning,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 115–123, Apr. 1987.
- [57] J.-J. Slotine and H. Yang, “Improving the efficiency of time-optimal path-following algorithms,” *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 118–124, Feb. 1989.
- [58] S. Singh and M. C. Leu, “Optimal Trajectory Generation for Robotic Manipulators Using Dynamic Programming,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 2, pp. 88–96, Jun. 1987. [Online]. Available: <https://doi.org/10.1115/1.3143842>
- [59] K. Hauser, “Fast Interpolation and Time-Optimization on Implicit Contact Submanifolds,” in *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, Jun. 2013. [Online]. Available: <http://www.roboticsproceedings.org/rss09/p22.pdf>
- [60] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, “Time-Optimal Path Tracking for Robots: A Convex Optimization Approach,” *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.
- [61] K. Shin and N. McKay, “A Dynamic Programming approach to trajectory planning of robotic manipulators,” *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491–500, Jun. 1986.
- [62] D. Constantinescu and E. A. Croft, “Smooth and time-optimal trajectory planning for industrial manipulators along specified paths,” *Journal of Robotic Systems*, vol. 17, no. 5, pp. 233–249, 2000. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4563%28200005%2917%3A5%3C233%3A%3AAID-ROB1%3E3.0.CO%3B2-Y>
- [63] Z. Shiller, “Time-energy optimal control of articulated systems with geometric path constraints,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 2680–2685 vol.4.
- [64] J. T. Betts and W. P. Huffman, “Path-constrained trajectory optimization using sparse sequential quadratic programming,” *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 1, pp. 59–68, 1993. [Online]. Available: <https://doi.org/10.2514/3.11428>
- [65] H. Pham and Q.-C. Pham, “A new approach to time-optimal path parameterization based on reachability analysis,” *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, 2018.
- [66] R.-J. Jing, M. Moreno-Maza, and D. Talaashrafi, “Complexity estimates for fourier-motzkin elimination,” in *Computer Algebra in Scientific Computing*, F. Boulier, M. England, T. M. Sadykov, and E. V. Vorozhtsov, Eds. Cham: Springer International Publishing, 2020, pp. 282–306.
- [67] T. Lipp and S. Boyd, “Minimum-time speed optimisation over a fixed path,” *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, Jun. 2014. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/00207179.2013.875224>
- [68] G. Chen, F. Dellaert, and S. Hutchinson, “Generalizing trajectory retiming to quadratic objective functions,” in *2024 International Conference on Robotics and Automation (ICRA)*, 2024 (in review). [Online]. Available: [https://gerry-chen.com/publications/Chen24icra\\_qopp/Chen24icra\\_qopp.pdf](https://gerry-chen.com/publications/Chen24icra_qopp/Chen24icra_qopp.pdf)
- [69] H. Pham and Q.-C. Pham, “A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis,” *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, Jun. 2018.
- [70] Z. Zou, T. Shi, S. Qiu, Y. Yuan, and Z. Shi, “Stylized neural painting,” 2020.
- [71] Z. Huang, W. Heng, and S. Zhou, “Learning to paint with model-based deep reinforcement learning,” in *Proceedings of*

- the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [72] Q. Wang, C. Guo, H.-N. Dai, and P. Li, “Stroke-gan painter: Learning to paint artworks using stroke-style generative adversarial networks,” *Computational Visual Media*, vol. 9, no. 4, pp. 787–806, 2023. [Online]. Available: <https://doi.org/10.1007/s41095-022-0287-3>
- [73] D. Ha and D. Eck, “A neural representation of sketch drawings,” in *ICLR 2018*, 2018, 2018. [Online]. Available: <https://openreview.net/pdf?id=Hy6GHpkCW>
- [74] Google, J. Jongejean, H. Rowley, T. Kawashima, J. Kim, and N. Fox-Gieg, “The Quick, Draw! dataset,” <https://github.com/googlecreativelab/quickdraw-dataset>, Google, 2023, the Quick Draw Dataset is a collection of 50 million drawings across 345 categories, contributed by players of the game “Quick, Draw!”. The drawings were captured as timestamped vectors, tagged with metadata including what the player was asked to draw and in which country the player was located. [Online]. Available: <https://github.com/googlecreativelab/quickdraw-dataset>
- [75] N. Jaques, J. McCleary, J. Engel, D. Ha, F. Bertsch, R. Picard, and D. Eck, “Learning via social awareness: Improving a deep generative sketching model with facial feedback,” 2018.
- [76] A. Das, Y. Yang, T. Hospedales, T. Xiang, and Y.-Z. Song, “Béziersketch: A generative model for scalable vector sketches,” 2020.
- [77] S. Ge, V. Goswami, C. L. Zitnick, and D. Parikh, “Creative sketch generation,” 2020.
- [78] A. Jenal, N. Savinov, T. Sattler, and G. Chaurasia, “Rnn-based generative model for fine-grained sketching,” 2019.
- [79] K. Frans, L. Soros, and O. Witkowski, “Clipdraw: Exploring text-to-drawing synthesis through language-image encoders,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 5207–5218. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/21f76686538a5f06dc431efea5f475f5-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/21f76686538a5f06dc431efea5f475f5-Paper-Conference.pdf)
- [80] P. Schaldenbrand, Z. Liu, and J. Oh, “Styleclipdraw: Coupling content and style in text-to-drawing translation,” 2022.
- [81] Y. Vinker, E. Pajouheshgar, J. Y. Bo, R. C. Bachmann, A. H. Bermano, D. Cohen-Or, A. Zamir, and A. Shamir, “Clipasso: Semantically-aware object sketching,” *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: <https://doi.org/10.1145/3528223.3530068>
- [82] Y. Vinker, Y. Alaluf, D. Cohen-Or, and A. Shamir, “Clipascene: Scene sketching with different types and levels of abstraction,” 2023.
- [83] T.-M. Li, M. Lukáč, M. Gharbi, and J. Ragan-Kelley, “Differentiable vector graphics rasterization for editing and learning,” *ACM Trans. Graph.*, vol. 39, no. 6, nov 2020. [Online]. Available: <https://doi.org/10.1145/3414685.3417871>
- [84] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” 2021.
- [85] Z. Qu, T. Xiang, and Y.-Z. Song, “Sketchdreamer: Interactive text-augmented creative sketch ideation,” 2023.
- [86] X. Xing, C. Wang, H. Zhou, J. Zhang, Q. Yu, and D. Xu, “Diffsketcher: Text guided vector sketch synthesis through latent diffusion models,” 2023.
- [87] Q. Wang, D. Kong, F. Lin, and Y. Qi, “Diffsketching: Sketch control image synthesis with diffusion models,” 2023.
- [88] P. Xu, T. M. Hospedales, Q. Yin, Y.-Z. Song, T. Xiang, and L. Wang, “Deep learning for free-hand sketch: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 285–312, 2023.
- [89] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, “Do as i can and not as i say: Grounding language in robotic affordances,” in *arXiv preprint arXiv:2204.01691*, 2022.
- [90] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, “Palm-e: An embodied multimodal language model,” in *arXiv preprint arXiv:2303.03378*, 2023.
- [91] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitzkovich, “Rt-1: Robotics transformer for real-world control at scale,” 2023.
- [92] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitzkovich, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” in *arXiv preprint arXiv:2307.15818*, 2023.
- [93] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, “Code as policies: Language model programs for embodied control,” in *arXiv preprint arXiv:2209.07753*, 2022.
- [94] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li, “Instruct2act: Mapping multi-modality instructions to robotic actions with large language model,” 2023.
- [95] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, “Progprompt: Generating situated robot task plans using large language models,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 523–11 530.
- [96] A. Bucker, L. Figueiredo, S. Haddadinl, A. Kapoor, S. Ma, and R. Bonatti, “Reshaping robot trajectories using natural language commands: A study of multi-modal data alignment using transformers,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 978–984.
- [97] A. Bucker, L. Figueiredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti, “Latte: Language trajectory transformer,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7287–7294.
- [98] Z. Wang, J. J. Hunt, and M. Zhou, “Diffusion policies as an expressive policy class for offline reinforcement learning,” 2023.
- [99] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, “Planning with diffusion for flexible behavior synthesis,” in *International Conference on Machine Learning*, 2022.
- [100] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, “Is conditional generative modeling all you need for decision making?” in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=sP1fo2K9DFG>
- [101] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S092188901300167X>
- [102] H. Choset, “Coverage for robotics – a survey of recent results,” *Annals of Mathematics and Artificial Intelligence*,

- vol. 31, no. 1, pp. 113–126, 2001. [Online]. Available: <https://doi.org/10.1023/A:1016639210559>
- [103] F. Devai, “Quadratic bounds for hidden line elimination,” in *Proceedings of the Second Annual Symposium on Computational Geometry*, ser. SCG ’86. New York, NY, USA: Association for Computing Machinery, 1986, pp. 269–275. [Online]. Available: <https://doi.org/10.1145/10515.10544>
- [104] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
- [105] Z. Kira, “Awesome-llm-robotics,” 2022. [Online]. Available: <https://github.com/GT-RIPL/Awesome-LLM-Robotics>
- [106] O. X.-E. Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu, C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns, F. Ceola, F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, H. Su, H.-S. Fang, H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radovicovic, I. Leal, J. Liang, J. Kim, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Wu, J. Luo, J. Gu, J. Tan, J. Oh, J. Malik, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Zhang, K. Majd, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Tan, L. Ott, L. Lee, M. Tomizuka, M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srivama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. D. Palo, N. M. M. Shafiuallah, O. Mees, O. Kroemer, P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundaresan, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Mendonça, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Moore, S. Bahl, S. Dass, S. Song, S. Xu, S. Haldar, S. Adeboala, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. hua Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Xu, and Z. J. Cui, “Open X-Embodiment: Robotic learning datasets and RT-X models,” <https://robotics-transformer-x.github.io>, 2023.
- [107] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. [Online]. Available: <https://openreview.net/forum?id=qw8AKxfYbI>
- [108] J. Wilkinson, E. Roth, T. Watson, C. Sugue, and T. Vandelin, “#000000book.com an open database for Graffiti Markup Language (GML) files.” <https://000000book.com/>, 2021.
- [109] OpenAI, “GPT-4V(ision) system card,” OpenAI, Tech. Rep., September 2023. [Online]. Available: [https://cdn.openai.com/papers/GPTV\\_System\\_Card.pdf](https://cdn.openai.com/papers/GPTV_System_Card.pdf)
- [110] H. Liu, C. Li, Y. Li, and Y. J. Lee, “Improved baselines with visual instruction tuning,” 2023.
- [111] A. Mandlekar, C. Garrett, D. Xu, and D. Fox, “Human-in-the-loop task and motion planning for imitation learning,” in *7th Annual Conference on Robot Learning*, 2023.
- [112] S. Kuhar, S. Cheng, S. Chopra, M. Bronars, and D. Xu, “Learning to discern: Imitating heterogeneous human demonstrations with preference and representation learning,” in *Conference on Robot Learning (CoRL)*, 2023.
- [113] L. A. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis using convolutional neural networks,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS’15. Cambridge, MA, USA: MIT Press, 2015, pp. 262–270.
- [114] V. Dumoulin, J. Shlens, and M. Kudlur, “A learned representation for artistic style,” in *International Conference on Learning Representations*, 2017. [Online]. Available: <https://openreview.net/forum?id=BJO-Bt1g>
- [115] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1510–1519.
- [116] M. L. Schrum, E. Hedlund-Botti, N. Moorman, and M. C. Gombolay, “Mind meld: Personalized meta-learning for robot-centric imitation learning,” in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI ’22. IEEE Press, 2022, pp. 157–165.
- [117] G. Chen, H. Muriki, A. Sharkey, C. Pradalier, Y. Chen, and F. Dellaert, “A hybrid cable-driven robot for non-destructive leafy plant monitoring and mass estimation using structure from motion,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 809–11 816. [Online]. Available: [https://gerry-chen.com/publications/Chen23icra\\_hydroponics\\_robot/Chen23icra\\_hydroponics\\_robot.pdf](https://gerry-chen.com/publications/Chen23icra_hydroponics_robot/Chen23icra_hydroponics_robot.pdf)
- [118] M. LiBretto, Y. H. Ahn, C. S. Han, Y. K. Cho, and J. Ueda, “Configuration Optimization for End-Point Stabilization of Redundant Manipulators With Base Flexibility,” *ASME Letters in Dynamic Systems and Control*, vol. 1, no. 2, p. 021001, 04 2020. [Online]. Available: <https://doi.org/10.1115/1.4046684>
- [119] G. Gongor, M. Turhan, H. Jamshidifar, and B. Fidan, “Online estimation and compensation of friction in industrial cable robot manipulation,” *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1332 – 1337, 2015, 15th IFAC Symposium onInformation Control Problems inManufacturing. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896315005091>
- [120] P. D. Campbell, P. L. Swaim, and C. J. Thompson, “Charlotte™ robot technology for space and terrestrial applications,” *SAE Transactions*, vol. 104, pp. 641–648, 1995. [Online]. Available: <http://www.jstor.org/stable/44611968>
- [121] M. A. Khosravi and H. D. Taghirad, “Dynamic modeling and control of parallel robots with elastic cables: Singular perturbation approach,” *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 694–704, 2014.
- [122] W. Shang, B. Zhang, B. Zhang, F. Zhang, and S. Cong, “Synchronization control in the cable space for cable-driven parallel robots,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4544–4554, 2019.
- [123] W. Shang, B. Zhang, S. Cong, and Y. Lou, “Dual-space adaptive synchronization control of redundantly-actuated cable-driven parallel robots,” *Mechanism and Machine Theory*, vol. 152, p. 103954, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094114X20301750>
- [124] J. Begey, L. Cuvillon, M. Lesellier, M. Gouttefarde, and J. Gangloff, “Dynamic control of parallel robots driven by flexible cables and actuated by position-controlled winches,” *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 286–293, 2019.
- [125] M. H. Korayem, M. Yousefzadeh, and B. Beyranvand, “Dynamics and control of a 6-dof cable-driven parallel robot with visco-elastic cables in presence of measurement noise,” *Journal of Intelligent & Robotic Systems*, vol. 88, no. 1, pp. 73–95, 2017. [Online]. Available: <https://doi.org/10.1007/s10846-017-0546-1>
- [126] M. Korayem, H. Tourajizadeh, and M. Bamdad, “Dynamic load carrying capacity of flexible cable suspended robot: Robust feedback linearization control approach,” *Journal of Intelligent and Robotic Systems*, vol. 60, pp. 341–363, 12 2010.
- [127] J. Niu, Q. Yang, G. Chen, and R. Song, “Nonlinear disturbance observer based sliding mode control of a cable-driven rehabilitation robot,” in *2017 International Conference on Rehabilitation Robotics (ICORR)*, 2017, pp. 664–669.
- [128] C. Schenk, H. H. Bülfhoff, and C. Masone, “Robust adaptive sliding mode control of a redundant cable driven parallel robot,” in *2015 19th International Conference on System Theory, Control and Computing (ICSTCC)*, 2015, pp. 427–434.
- [129] C. Schenk, C. Masone, A. Pott, and H. H. Bülfhoff, “Application of a differentiator-based adaptive super-twisting controller for a redundant cable-driven parallel robot,” in *Cable-Driven Parallel*

- Robots*, C. Gosselin, P. Cardou, T. Bruckmann, and A. Pott, Eds. Cham: Springer International Publishing, 2018, pp. 254–267.
- [130] *Design and Application of Chattering-Free Sliding Mode Controller to Cable-Driven Parallel Robot Manipulator: Theory and Experiment*, ser. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. Volume 2: 34th Annual Mechanisms and Robotics Conference, Parts A and B, 08 2010. [Online]. Available: <https://doi.org/10.1115/DETC2010-29153>
- [131] Y. Wang, J. Chen, F. Yan, K. Zhu, and B. Chen, “Adaptive super-twisting fractional-order nonsingular terminal sliding mode control of cable-driven manipulators,” *ISA Transactions*, vol. 86, pp. 163–180, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0019057818304440>
- [132] E. Picard, E. Tahoumi, F. Plestan, S. Caro, and F. Claveau, “A new control scheme of cable-driven parallel robot balancing between sliding mode and linear feedback,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9936–9943, 2020, 21st IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896320334716>
- [133] M. H. Korayem, H. Tourajizadeh, M. Jalali, and E. Omidi, “Optimal path planning of spatial cable robot using optimal sliding mode control,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 5, p. 168, 2012. [Online]. Available: <https://doi.org/10.5772/52892>
- [134] C. Lambert, M. Nahon, and D. Chalmers, “Implementation of an aerostat positioning system with cable control,” *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 1, pp. 32–40, 2007.
- [135] J. C. Santos, A. Chemori, and M. Gouttefarde, “Model predictive control of large-dimension cable-driven parallel robots,” in *Cable-Driven Parallel Robots*, A. Pott and T. Bruckmann, Eds. Cham: Springer International Publishing, 2019, pp. 221–232.
- [136] J. C. Santos, M. Gouttefarde, and A. Chemori, “A nonlinear model predictive control for the position tracking of cable-driven parallel robots,” *IEEE Transactions on Robotics*, pp. 1–20, 2022.
- [137] C. Song and D. Lau, “Workspace-based model predictive control for cable-driven robots,” *IEEE Transactions on Robotics*, pp. 1–20, 2022.
- [138] M. Katliar, J. Fischer, G. Frison, M. Diehl, H. Teufel, and H. H. Bülfhoff, “Nonlinear model predictive control of a cable-robot-based motion simulator,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9833–9839, 2017, 20th IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896317313678>
- [139] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
- [140] A. Pott, T. Bruckmann, and L. Mikelsons, “Closed-form force distribution for parallel wire robots,” in *Computational Kinematics*, A. Kecskeméthy and A. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 25–34.
- [141] F. Dellaert *et al.* (2022) Georgia Tech smoothing and mapping (GTSAM). <https://github.com/borglab/gtsam>.
- [142] E. Todorov, “Optimal control theory,” in *Bayesian Brain: Probabilistic Approaches to Neural Coding*. MIT Press, 2006, pp. 269–298. [Online]. Available: [https://courses.cs.washington.edu/courses/cse528/09sp/optimality\\_chapter.pdf](https://courses.cs.washington.edu/courses/cse528/09sp/optimality_chapter.pdf)

## APPENDIX A CABLE ROBOT ADDITIONAL RELATED WORKS

In this appendix we describe related works for CDPR state estimation and tracking control in more detail. Further reading can also be found in [3], [27], [49]

### *CDPR State Estimation Related Works*

Given existing calibration parameters  $\Theta$ , pose estimation is the forward kinematics problem of estimating end-effector pose from motor joint angles. As compared to serial robots where forward kinematics (FK) has a closed-form expression while inverse kinematics (IK) can be computed using numerical optimization, CDPRs (which are parallel robots) are exactly the opposite: IK has a closed-form expression while FK is most often computed by solving a nonlinear least squares (NLS) problem online, of the form  $T^* = \arg \min_T \|IK(T; \Theta) - l(\theta; \Theta)\|^2$  where  $T^*$  is the estimated pose,  $IK$  computes the inverse kinematics, and  $l$  computes the cable lengths given the motor angles  $\theta$ . Although this NLS problem can be solved very efficiently using Gauss-Newton iteration in fewer than 10 steps even with a very naïve initial estimate for  $T$  [27], there can occasionally be multiple local minima representing multiple valid poses for the same cable lengths [27] (analogous to redundancy resolution in serial robots).

Offline self-calibration, in which we seek to compute kinematic, statics, and dynamic parameters, is largely divided into perturbation [28], [29] vs passive measurement [30], [31] techniques. The kinematic parameters include (a) affine, linear (winch radius), and nonlinear parameters in the mapping of motor angle to cable length, and (b) cable attachment points on the static frame and moving end-effector. The static and dynamic parameters include cable stiffness/damping, friction coefficients, and motor and end-effector inertias. Perturbation techniques involve making small cable length or tension perturbations one cable at a time and measuring the responses in the other cable lengths/tensions to measure a numerical approximation to the Jacobian at that configuration. Depending on the set of parameters known vs sought, the procedure can be repeated at other configurations as well to obtain the kinematic parameters. Passive measurement techniques involve passively collecting the cable lengths and tensions while the robot is moved, and solving an optimization problem to compute the parameters, since the measurements are constrained by equations conditioned on kinematic, static, and dynamic parameters. Perturbation techniques can be dangerous and obtrusive to the application of the cable robot, since the robot needs to perform set motions before it is calibrated which may interfere with the application *e.g.* through collision or interrupting motions when recalibration is needed. Passive measurement techniques allow calibration to

be performed while the robot is performing its usual motions for the application, but may be less accurate and requires additional considerations to be applied to online, continuous calibration.

Online continuous calibration [119] is desireable for long-term and robust operation of CDPRs, especially since cable properties can drift with environmental conditions and wear, and winch winding parameters may not be tightly controlled in low-cost winch designs. Online continuous calibration is implemented as an extension of passive measurement calibration, in which the passive calibration is periodically re-done on fixed intervals or when a calibration quality metric threshold is reached, or in which the parameters are continuously updated using a Bayesian filter. Both suffer from the same issues that diverse measurements across the workspace is needed to avoid poorly conditioned data, so special considerations are needed to ensure that sufficiently diverse measurements are collected and saved.

### *CDPR Tracking Control Related Works*

Tracking control for CDPRs has been well studied since at least the late 1980's and has progressed from simple joint-space methods to advanced nonlinear, adaptive, and robust methods. The earliest examples of CDPR tracking control commanded position-controlled servos to the cable lengths computed by inverse kinematics, but these require tedious, critical calibration and are sensitive to modelling errors [120]. More advanced model-based techniques include operational-space PID, feedback linearization, sliding-mode control (SMC), linear quadratic (LQR and LQG), and MPC, among others [121]. Linear approaches (*e.g.* PID) with feedforward terms to handle most of the nonlinearity are perhaps the most common [32], [41], [122], [123], but the use of static gains can make them sensitive to operating point, leading to non-uniform performance across the workspace and lack of foresight when approaching control limits [34]. Feedback linearization approaches have seen success in *suspended* cable robot designs but require special considerations for redundantly actuated CDPR [34], [124]–[126]. Feedback linearization may also face difficulties modeling control limits. SMC [33] is perhaps most popular nonlinear approach for robust CDPR control, but is prone to chattering [127] due to control discontinuities. Although reducing chattering issues has been studied by several works [128]–[132], chattering does not appear to be completely resolved and these methods rely on adaptive algorithms to fallback onto linear controllers which, again, may not be workspace-aware. LQR and LQG have been primarily applied as tuning techniques for other controllers (*e.g.* tuning SMC [133] or feedback linearization gains [125]) and therefore inherit their limitations. Furthermore, applying LQR

and LQG to systems that have been “pre-linearized” by other approaches tend to use fixed gains obtained using the infinite horizon LQR solution [133], [134]. Linear [135] and Nonlinear MPC [136]–[138] are highly capable in providing consistent and robust performance near or even outside workspace boundaries [34], but MPC poses implementation challenges and must often make convexifying approximations to guarantee online convergence [34].

Ideally, a control scheme would combine the accuracy and robustness of MPC approaches with the simplicity and ease of implementation of linear approaches.

## APPENDIX B

### CABLE ROBOT CONTROL AND ESTIMATION DETAILS

In this appendix we describe additional mathematical details about CDPR state estimation and control using factor graphs.

We will attempt to describe the most pertinent concepts about factor graphs as we go, but additional introductions to factor graphs can be found in [46], [139].

#### A. Trajectory Generation

We seek to track the reference trajectory while also maintaining realizable control inputs. Least-squares objectives are used to minimize both the tracking error,  $\tilde{x} := x - {}^d x$ , and the control input objective,  $\tilde{u} := u - u_{mid}$ , where  ${}^d x$  is the desired trajectory and  $u_{mid} := \frac{1}{2}(u_{min} + u_{max})$  as described by [140] as an approximation for maximizing the margin to control limits. Constraints are given by the system dynamics. Our trajectory generation problem can then be expressed as in Fig. 13, which graphically depicts the trajectory generation problem as both a factor graph and the equivalent constrained nonlinear least squares problem: eq. (21). The pose and twist,  $T, \mathcal{V}$ , make up  $x$  and the tensions,  $\tau_0, \tau_1, \tau_2, \tau_3$ , make up  $u$ .

We use the software library Georgia Tech Smoothing and Mapping (GTSAM) [141] to create and “solve” our factor graph, where “solve” is used to mean computing the solution to the equivalent constrained least squares problem. We use the Levenberg-Marquardt optimizer with variable elimination for the inner-loop linear solver (default solver in GTSAM). It can be shown that Levenberg-Marquardt with variable elimination is identical to iLQR as described by [48], although we run this optimization offline. As a result, feasibility (incl. control limits) and optimality can be checked prior to executing on the robot.

The computed solution to the factor graph is the nominal, feedforward trajectory and denoted  $x^*, u^*$ .

#### B. Control

We use a time varying linear-quadratic regular (LQR) for our controller which is computed by linearizing the system around  $x^*, u^*$  and applying variable elimination.

*a) Linearizing the graph:* The linearization process is automatically handled by GTSAM, but mathematically it equates to applying a first-order Taylor expansion around  $x^*, u^*$  to convert the general optimal control problem (21) (which is transcribed as a constrained *nonlinear* least squares optimization) into the LQR problem (22) (which is transcribed as a constrained *linear* least squares problem). Upon linearizing, we will define  $\delta x_k := x_k - x_k^*$  and  $\delta u_k := u_k - u_k^*$ .

The result of linearizing is a graph with the same structure as in Fig. 13 except with all the factors representing either linear constraints or quadratic objectives.

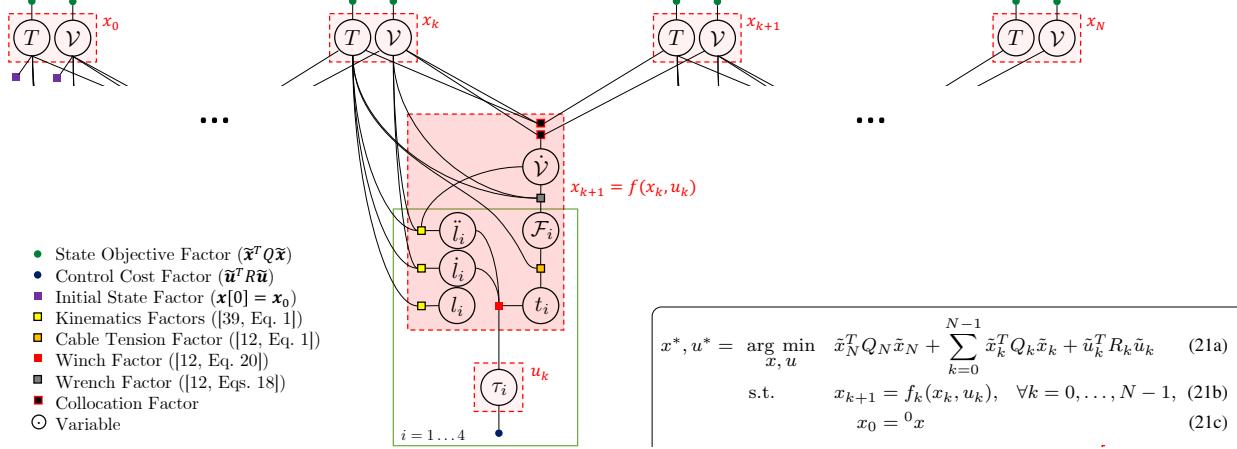


Fig. 13. Factor graph describing the full cable robot kinematics, dynamics, initial state, and state/control objectives, together with the equivalent optimization problem description (lower right). The kinematics and dynamics constraints, which are represented by square factors and make up (21b):  $x_{k+1} = f_k(x_k, u_k)$ , are defined according to standard cable robot literature. For example, the orange cable tension factor encodes the constraint  $\mathbf{W}t = \mathcal{F}$  where  $\mathbf{W}$  is the wrench matrix computed from the pose  $T$ . Example references are given in the legend.  $T$  denotes the pose of the end effector;  $V$  denotes the twist of the end effector;  $\dot{V}$  denotes the twist acceleration of the end effector;  $l_i, \dot{l}_i, \ddot{l}_i, t_i$  denote the length, velocity, acceleration, and tension of cable  $i$  respectively;  $\tau_i$  denotes the torque commanded for motor  $i$ ; and  $\mathcal{F}_i$  denotes the wrench applied to the end effector by cable  $i$ . Time indices are omitted for notational clarity. Together,  $T, V$  make up the state  $x$  and the torques  $\tau_0, \dots, \tau_3$  make up the control  $u$ .

Mathematically, the linearized graph now represents a *linear* least squares problem (with linear equality constraints) which can be efficiently and exactly solved using variable elimination (which is equivalent to standard sparse linear algebra algorithms).

*b) Variable Elimination:* We first give a brief introduction to the variable elimination algorithm, then show how it can be applied to extract LQR gains from our linearized graph.

The *variable elimination algorithm* is nothing more than an algorithmic method of solving for a variable and substituting the solution expression back into the system of equations. Although GTSAM handles elimination for us, we can briefly summarize its underlying computation for linear problems as performing the block QR factorization on a subsystem:

$$\begin{bmatrix} M_{11} & M_{12} & b_1 \\ M_{21} & M_{22} & b_2 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = Q \begin{bmatrix} R_{11} & S_{12} & d_1 \\ 0 & A_z & d_2 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}$$

where  $M, b$  describes a linear least squares problem,  $y$  are the variables we seek to eliminate,  $z$  are the remaining variables (formally,  $z$  is the *separator*: set of variables not in  $y$  that share a factor with any of the variables in  $y$ ), and  $Q, R, S, A, d$  are results of the factorization. The top row encodes the solution  $y = R_{11}^{-1}(d_1 - S_{12}z)$  and the bottom row encodes the remaining system after substituting the solution for  $y$  back in:  $A_z z = d_2$ . In other words, factorization is equivalent to solving for a variable and substituting it back into the original system. A comprehensive description of the variable elimination algorithm can be found in [46] for interested readers.

From the linearized graph, we first apply variable elimination to eliminate all variables except  $\delta x, \delta u$ , meaning we eliminate  $y = \{\delta l, \delta \dot{l}, \delta \ddot{l}, \delta t, \delta \mathcal{F}, \delta \dot{V}\}$  but leave  $z = \{\delta T, \delta V, \delta \tau\}$  (for every timestep). Reiterating, this is equivalent to solving for  $y$  first and substituting back into (22). After eliminating  $y$ , the resulting graph/system for  $z$  will have the form of Fig. 14 which can be equivalently described by the linear least squares problem (22). Also note that the computer has performed all the algebraic manipulation for us. Introducing *e.g.* additional parameters, periodicity constraint factors, and state-dependent control limits are all possible with minimal designer effort.

Finally, we can also use variable elimination to derive the well known result of linear-quadratic control [142] that the solution  $\delta u^*(\delta x)$  can be expressed as the linear control law:

$$\delta u_k^*(x_k) = K_k \delta x_k. \quad (25)$$

Notice that, since  $x^*, u^*$  satisfy (21), they are also guaranteed to satisfy (22). Thus when  $\delta x_k = 0$ ,  $\delta u_k^*(0) = 0$  so the affine term is zero.

To obtain the LQR gains,  $K_k$ , from the linear graph in Fig. 14, we eliminate the states and controls one at a time in the order  $\delta x_N, \delta u_{N-1}, \delta x_{N-1}, \dots, \delta u_0, \delta x_0$ . It was proved in [44], [45] that this variable elimination process produces the finite horizon discrete LQR solution. As a brief intuition for the proof, this elimination follows the standard logic (backward induction on the Bellman equation) for deriving the finite horizon discrete algebraic riccati equation: given the value function for  $x_{k+1}$ ,

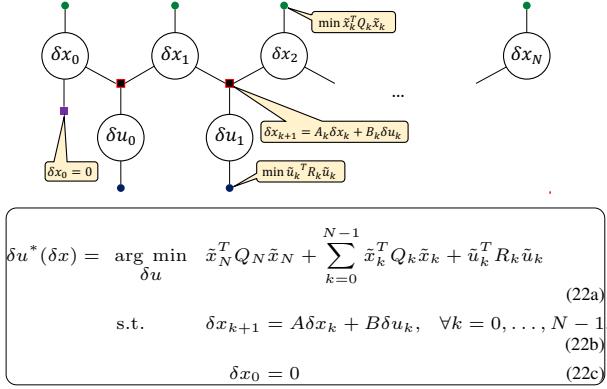


Fig. 14. After *linearizing* Fig. 13 and applying *variable elimination* to eliminate all the variables that are neither state nor control, we obtain this LQR graph (almost identical to Fig. 4), which can be equivalently represented as (22). Green and navy blue factors represent state and control costs respectively, red-outlined-black factors represent dynamics constraints, and the purple factor represents the initial state constraint.

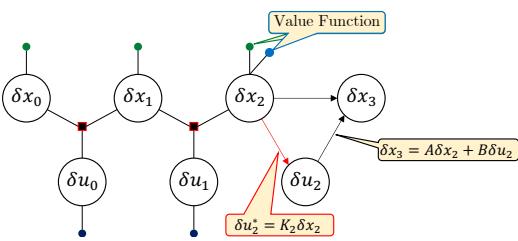


Fig. 15. Variable elimination is used to compute the optimal time-varying feedback control law starting at the final timestep and eliminating back to the beginning. This diagram depicts eliminating two variables:  $\delta x_3, \delta u_3$ . The blue factor is the cost that gets propagated upon “substituting” the optimal value back into the system, which makes up the value function.

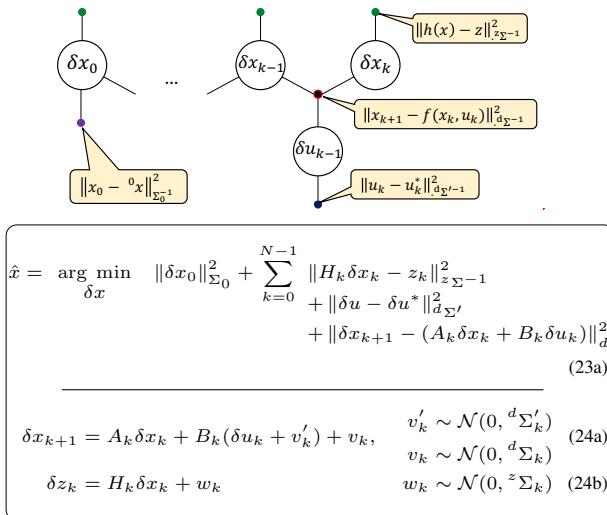


Fig. 16. Nonlinear estimation factor graph (top) and the linearized estimation problem expressed as a minimization problem (middle) and system (bottom). After linearizing, we introduce the variables  $\delta z_k := z - z^*$  and  $z^* := h(x_k^*)$ , and  $H_k := \frac{\partial h_k}{\partial x_k} \Big|_{x_k^*}$ .

solve for the  $u_k$  which minimizes the value function then substitute  $u_k^*$  back into the value function to obtain the new value function for  $x_k$ . Fig. 15 visually illustrates the elimination process.

### C. Estimation

We can express our estimation problem by converting the factor graph in Fig. 13 to replace controls factors by estimation factors, followed by linearization and marginalization.

*a) Estimation Factor Graph:* To convert from the optimal control factor graph to the estimation factor graph, we make the following factor replacements:

$\tilde{x}_k^T Q_k \tilde{x}_k$ (state cost)	$\ h_k(x_k) - z_k\ _{z\Sigma_k^{-1}}^2$ (stochastic measurement)
$x_{k+1} = f(x_k, u_k)$ (dyn. constr.)	$\ x_{k+1} - f(x_k, u_k)\ _{d\Sigma_k^{-1}}^2$ (modelling uncertainty)
$\tilde{u}_k^T R_k \tilde{u}_k$ (control cost)	$\ u_k - u_k^*\ _{d\Sigma_k'^{-1}}^2$ (input disturbance)

where  $\|\square\|_{\Sigma^{-1}} := \square^T \Sigma^{-1} \square$  denotes the Mahalanobis norm;  $z_k := [l_k, \dot{l}_k]$  denotes the measurement vector at timestep  $k$ ;  $u_k^*$  and  $u_k$  denote the commanded and realized control torques vectors respectively;  $h(x_k)$  denotes the inverse kinematics to compute the cable length given the state; and  ${}^z\Sigma$ ,  ${}^d\Sigma$ ,  ${}^d\Sigma'$  denote the measurement, dynamics, and input covariances respectively.

The factor graph now represents the system:

$$x_{k+1} = f(x_k, u_k + v'_k) + v_k, \quad v'_k \sim \mathcal{N}(0, {}^d\Sigma'_k) \quad (26)$$

$$v_k \sim \mathcal{N}(0, {}^d\Sigma_k) \quad (27)$$

$$z_k = h(x_k) + w_k \quad w_k \sim \mathcal{N}(0, {}^z\Sigma_k) \quad (28)$$

*b) Linearization:* Next, we linearize and eliminate the intermediate variables the same way as in Section B-B to obtain the factor graph shown in Fig. 16 which is equivalent to the system (24) in Fig. 16.

After linearizing,  $v'_k$  can be combined with  $v_k$  by applying the transformation:

$$\delta x_{k+1} = A_k \delta x_k + B_k (\delta u_k + v'_k) \quad (29)$$

$$= A_k \delta x_k + B_k \delta u_k + B_k v'_k \quad (30)$$

where  $v'_k \sim \mathcal{N}(0, {}^d\Sigma')$  so  $v_k \sim \mathcal{N}(0, (B_k^T {}^d\Sigma'^{-1} B_k)^{-1})$  and  ${}^d\Sigma_k = (B_k^T {}^d\Sigma'^{-1} B_k)^{-1}$ . This is useful because most disturbances in our CDPR setup come in the form of parasitic torques/tensions in the cable as opposed to external wrenches acting directly on the end effector. Meanwhile, removing  $v'_k$  simplifies the factor graph and the notation.

c) *Marginalization to Extract Kalman Gains:* We now seek a Kalman Filter to estimate  $\hat{x}_k = E[\delta x_k]$ . Specifically, we would like to pre-compute the Kalman Gains for this filter to avoid needing to compute them online. Although, to the best of our knowledge, factor graphs cannot directly extract the Kalman Gains the way we were able to extract the LQR gains, they can nevertheless get us most of the way and provide intuition for the final expression.

First, we can use marginalization to compute all the *a priori* and *a posteriori* covariances. In these cases, marginalization can be computed by simply eliminating every variable other than the one we seek the marginal of. Fig. 17 and 18 show how variable elimination is used to compute the new covariances after predicting and updating respectively. This computation is handled by GTSAM offline and  $\Sigma_{1|0}, \Sigma_1, \dots, \Sigma_{N|N-1}, \Sigma_N$  are the only results we need for the next step.

We now obtain expressions to compute  $\hat{x}$  given the covariance matrices we just computed. The equation to predict  $\hat{x}_{k|k-1}$  is straightforward since all noises are zero-mean:

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1} + B_k \delta u_{k-1}. \quad (31)$$

The equation to update  $\hat{x}_k$  is less obvious, but more intuitive by observing Fig. 18: the optimal estimate for  $\delta x_k$  in Fig. 18 is the one which makes the best compromise between  $\|\delta x_k - \hat{x}_{k|k-1}\|_{\Sigma_{k|k-1}^{-1}}$  and  $\|H_k \delta x_k - \delta z_k\|_{z \Sigma_k^{-1}}$ . We expect the optimal choice of  $\delta x_k$  to be a weighted mean of the two factors, and indeed with a bit of algebra it can be proven that it is:

$$\begin{aligned} \|\delta x_k - \hat{x}_k\|_{\Sigma_k^{-1}}^2 &= \|\delta x_k - \hat{x}_{k|k-1}\|_{\Sigma_{k|k-1}^{-1}}^2 \\ &\quad + \|H_k \delta x_k - \delta z_k\|_{z \Sigma_k^{-1}}^2 \\ \delta x_k^T \Sigma_k^{-1} \hat{x}_k &= \delta x_k^T \Sigma_{k|k-1}^{-1} \hat{x}_{k|k-1} + \delta x_k^T H_k^T z \Sigma_k^{-1} \delta z_k \\ \Sigma_k^{-1} \hat{x}_k &= \Sigma_{k|k-1}^{-1} \hat{x}_{k|k-1} + H_k^T z \Sigma_k^{-1} \delta z_k \\ \hat{x}_k &= \Sigma_k \left( \Sigma_{k|k-1}^{-1} \hat{x}_{k|k-1} + H_k^T z \Sigma_k^{-1} \delta z_k \right) \end{aligned} \quad (32)$$

Finally, the predict and update steps can be combined to form a single state estimator update equation:

$$\hat{x}_k = {}^x K_k \hat{x}_{k-1} + {}^u K_k \delta u_{k-1} + {}^z K_k \delta z_k \quad (33)$$

where  ${}^x K_k := \Sigma_k \Sigma_{k|k-1}^{-1/2} A_k$ ,  ${}^u K_k := \Sigma_k \Sigma_{k|k-1}^{-1/2} B_k$ , and  ${}^z K_k := \Sigma_k H_k^T z \Sigma_k^{-1/2}$  are pre-computed offline.

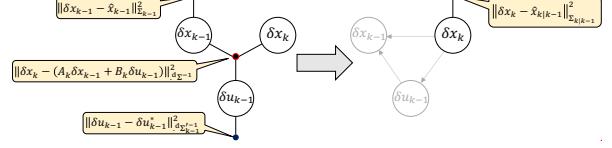


Fig. 17. Kalman Filter ‘‘Predict’’ step eliminates  $\delta x_{k-1}, \delta u_{k-1}$  to obtain  $\Sigma_{k|k-1}$ .

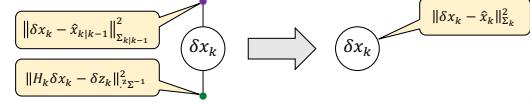


Fig. 18. Kalman Filter ‘‘Update’’ step eliminates  $\delta x_k$  to obtain  $\Sigma_k$ .

## APPENDIX C EXPRESSING CABLE ROBOT CONSTRAINTS IN THE TASK SPACE

To convert joint-space constraints to task-space constraints, we can leverage the kinematics of our cable driven parallel robot. Whereas serial manipulator kinematics are defined by  $\dot{\mathbf{x}} = J\dot{\mathbf{q}}$  so transforming task-space constraints to joint-space is straightforward, cable robot kinematics are defined by  $\dot{\mathbf{q}} = \mathbf{W}^T \dot{\mathbf{x}}$  (or equivalently  $\mathcal{F} = \mathbf{W}\mathbf{t}$ ) where  $\mathbf{W}$  is the wrench matrix [140] so transforming joint-space constraints to task-space is straightforward. The procedure is outlined as follows. The inverse kinematics for our CDPR are given by the closed-form expression  $\mathbf{q}(s) = \|\mathbf{T}\mathbf{b} - \mathbf{a}\|$  where  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{3 \times n}$  are the attachment points of the  $n$  cables to the robot base (in the world frame) and end-effector (in the end-effector frame) respectively and  $\mathbf{T}\mathbf{b}$  denotes coordinate transformation. Then,  $\dot{\mathbf{q}} = \mathbf{W}(\mathbf{T})^T \dot{\mathbf{x}}$  and  $\ddot{\mathbf{q}} = \mathbf{W}(\mathbf{T})^T \ddot{\mathbf{x}}$  where  $\mathbf{W}_i(\mathbf{T}) = [\mathbf{u}_i; \mathbf{b}_i \times \mathbf{u}_i]$  is the  $i^{th}$  column of the *wrench matrix* [140].  $\dot{\mathbf{q}}(s)$  is a linear function of  $\dot{\mathbf{x}}(s)$  and easy to compute analytically given  $\mathbf{T}$ , so clearly motor speed constraints can be expressed in the form of (7c). Finally, the equations of motion and torque constraints for the cable robot are given by

$$\mathbf{M}(s)\ddot{\mathbf{x}} + \dot{\mathbf{x}}^T \mathbf{C}(s)\dot{\mathbf{x}} + \mathbf{g}(s) = \mathcal{F} = \mathbf{W}(s)\mathbf{t} \quad (34)$$

$$I_i \ddot{q}_i + F_i(\dot{q}_i) \dot{q}_i + R t_i = \tau_i \quad (35)$$

$$\boldsymbol{\tau} \in \mathcal{C}^\tau. \quad (36)$$

Substituting  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  into (35) then solving for  $t_i$  to substitute into (34) often yields a constraint in the form of (7b). While  $F_i(\dot{q}_i)\dot{q}_i$  cannot *always* be turned into the quadratic form of (7b), common forms can be handled. For example, static friction modeled as  $f_s \text{sign}(\dot{q}_i)$  turns into a constant since we know a priori the path  $q(s)$  and therefore also the sign of  $\dot{q}_i$ . Viscous friction, approximated  $f_v \dot{q}_i$ , will introduce a linear term in  $\dot{\mathbf{x}}$ , but by completing the square (requires inverting  $\mathbf{C}(\mathbf{x})$ ) and reparameterizing  $\dot{\mathbf{x}}' := \dot{\mathbf{x}} + \frac{f_v}{R} \mathbf{C}^{-1} \mathbf{W} \mathbf{W}^T$ , this can also be turned into the form of (7b) taking care to carry this forward oneself for the remainder of the derivation.

## APPENDIX D FACTOR GRAPH LP SOLVING DETAILS

As a brief background on factor graph elimination, an optimization problem can be described with a factor graph by denoting each of the variables to be optimized as a “variable” node and each of the optimization objective terms and constraints as a “factor” node, where an edge connects each variable a factor depends on. Then to solve the optimization problem, the variable elimination algorithm states to “eliminate” (solve) one variable,  $x$ , at a time, passing its constraints and objective terms as a new factor on the *separator*,  $\mathcal{S}(x)$ : the set of all variables sharing a factor with the eliminated variable. A more complete description of factor graph elimination for solving optimal control problems can be found in [45], [49].

The factor graph for this problem is given in Fig. 7. As it applies to this problem, factor graph elimination would proceed eliminating one variable at a time in the order  $u_0, x_0, u_1, x_1, \dots, x_{N-1}, x_N$ . For example, following the *Reachable* set elimination ordering, we would first eliminate  $u_0$  by solving the following LP:

$$\begin{aligned} u_0^*(x_0, x_1) = & \underset{u_0}{\text{maximize}} && (\text{nothing}) \\ & \text{subject to} && \mathbf{a}_0 u_0 + \mathbf{b}_0 x_0 + \mathbf{c}_0 \in \mathcal{C}_0, \\ & && x_1 - x_0 - 2u_0 \Delta s = 0 \end{aligned}$$

In this case, although we have no objective function, the solution is obvious because the dynamics fully constrains  $u_0$ :

$$u_0^*(x_0, x_1) = \frac{1}{2\Delta s} (x_1 - x_0). \quad (37)$$

We then substitute  $u_0^*$  to create a new factor on the separator  $\mathcal{S}(u_0) = \{x_0, x_1\}$ :

$$\frac{\mathbf{a}_0}{2\Delta s} (x_1 - x_0) + \mathbf{b}_0 x_0 + \mathbf{c}_0 \in \mathcal{C}_0.$$

Next, we eliminate  $x_0$  in much the same way:

$$x_0^*(x_1) = \max_{x_0} x_0 \quad (38a)$$

$$\text{s.t. } \frac{\mathbf{a}_0}{2\Delta s} (x_1 - x_0) + \mathbf{b}_0 x_0 + \mathbf{c}_0 \in \mathcal{C}_0, \quad (38b)$$

$$x_0 > 0. \quad (38c)$$

For the purposes of variable elimination, consider that we don’t actually need to symbolically solve this LP, but instead we just need 2 things:

- 1) the optimal value of  $x_0$  as a function of  $x_1$ , and
- 2) the resulting objectives/constraints on  $x_1$  after we substitute  $x_0 = x_0^*(x_1)$ .

For 1, we do not need an analytical expression (yet) so we just store the conditional as the optimization problem (38).

For 2, the new factor will consist of an objective component and a constraint component.

The objective component is easy: we can ignore it because we will select  $x_1$  greedily. More formally, our new objective factor will be  $x_0^*(x_1)$ , but because we also have a pre-existing factor  $wx_1$  where  $w$  is a very large number, our new objective term is negligible in comparison ( $x_0^*(x_1) \ll wx_1$ ).

The constraint component of our new factor can be solved with 2 LPs the same way as in [69]. Since  $x_1$  is a scalar, the resulting constraint on  $x_1$  will take the form:

$$x_{1,min} \leq x_1 \leq x_{1,max}. \quad (39)$$

We compute the smallest and largest possible values of  $x_1$  that satisfy (38b), (38c):

$$x_{1,min} = \underset{x_0, x_1}{\text{minimize}} \quad x_1 \quad \text{subject to (38b), (38c)}, \quad (40)$$

$$x_{1,max} = \underset{x_0, x_1}{\text{maximize}} \quad x_1 \quad \text{subject to (38b), (38c)}. \quad (41)$$

These are very easy to solve in just a few dozen lines of code since we need only optimize over 2, scalar variables.

After eliminating  $u_0, x_0$  we have Fig. 8. We repeat the elimination process on  $u_1, x_1, \dots, x_{N-1}, x_N$  until all variables are eliminated. The result is the *Bayes Net* in Fig. 9.

*Back-substitution:* We solve the Bayes Net by back-substitution. The final elimination step will produce a *marginal* on  $x_N$  of the form:

$$\begin{aligned} x_N^* &= \underset{x_N}{\text{max}} \quad x_N \\ \text{s.t.} \quad x_{N,min} &\leq x_N \leq x_{N,max}, \\ x_N &> 0 \end{aligned} \quad (42)$$

whose solution is clearly  $x_N^* = x_{N,max}$ .

Then, we can compute  $x_{N-1}^* = x_{N-1}^*(x_N)$  by substituting  $x_N \leftarrow x_N^*$  into the conditional (19) and solving the now single-variable scalar LP (which is just iterating through the inequalities to find the lower bound) for  $x_{N-1}$ . This process is repeated until all variables are evaluated, and the resulting sequence  $x_0^*, \dots, x_N^*, u_0^*, \dots, u_{N-1}^*$  is the solution to (17).