

Direct System Identification of Deformable Objects using Differentiable Finite Element Dynamics

David Millard, James A. Preiss, Jernej Barbič, Gaurav S. Sukhatme

University of Southern California

Los Angeles CA 90089, USA

{dmillard, japreiss, jnb, gaurav}@usc.edu

Abstract—We consider the problem of identifying material parameters of a deformable object, such as elastic moduli, by non-destructive robotic manipulation. We assume known geometry and mass, a reliable fixed grasp, and the ability to track the positions of a few points on the object surface. We collect a dataset of grasp pose sequences and corresponding point position sequences. We represent the object by a tetrahedral Finite Element Method (FEM) mesh and optimize the material parameters to minimize the difference between the real and predicted observations. We use a collocation-type formulation where the sequence of FEM mesh states are decision variables, and the dynamics are encoded as constraints. Sparsity patterns in the constraints make this problem tractable despite the large number of variables. We hope that a collocation-based approach will increase the set of initial parameter guesses that converge to optimal system identifications. Preliminary experiments show that our approach is computationally feasible but that our penalty-based optimization algorithm may not enforce the dynamics constraints stringently enough, motivating ongoing inquiry into the most appropriate methods for our proposed optimization problem.

Index Terms—dynamical systems, parameter estimation, deformable objects

I. INTRODUCTION

Robotic manipulation of deformable objects is a valuable and challenging domain. Many objects of human necessity, like clothes, food, and bedding, move in ways which are dominated by their deformability. Systems capable of safe and reliable dexterous deformable manipulation would greatly increase the number of industrial, domestic, and logistical contexts in which robots could be deployed.

Deformable dynamics are difficult to model, manipulate, and even to perceive. Unlike many tasks with rigid objects, grasping in different locations on the same object will result in significantly different behavior. An object may fold to obscure itself or otherwise change shape dramatically, preventing a sensor system from reliable tracking over time.

A motivating example for system identification in robotics is the task of tracking a given fast trajectory with the distal end of a deformable object. The control task is dominated by the passive deformable dynamics of the object, and require accurate system parameter identification for model-based control. An example physical setup is shown in Figure 1.

In this work, we propose a method to determine the material parameters of a deformable object model, using data collected from robot interactions. In contrast to many common techniques



Fig. 1. A motivating tracking task, where a robot controller needs to manipulate a deformable object along a given trajectory, while predicting and controlling for the passive dynamic objects.

which require specialized force measurement equipment, our proposed method relies only on observations of the bulk motion of the object over time. We describe a nonlinear optimization problem formulation inspired by collocation methods, a penalty augmented loss transformation which enables optimization algorithms to solve it, and discuss sparse matrix structures which enable efficient solutions. We also present preliminary experiments which show that this method is promising for complex systems with deformable dynamics.

II. BACKGROUND AND RELATED WORKS

A. Simulation

Rich simulation environments have become a fundamental technology in autonomous robotics. An early and ongoing use of physics simulation is as a virtual testbed for algorithm and mechanism design [15]. In such uses, simulated environments provide a safe, inexpensive, fast, and easily configurable approximation of the world, and are primarily useful for decreasing engineering time and cost. The physical phenomena simulated by such systems can be complex and span many

domains within one system, including electrical, mechanical, thermal, with interactions between such subsystems [8].

With the advent of deep Reinforcement Learning (RL) techniques for robotics applications, simulators also play an important role in data collection during the learning process [1]. In this context, simulators again provide a safe environment for algorithms to execute suboptimal actions, which could be dangerous or damaging in the real world. The popular class of model-free RL algorithms require vast amounts of data from environment interactions to train [17], and simulators provide an inexpensive, safe, and often much faster avenue towards such data generation. Successful deep RL methods for dynamically manipulating deformable objects have been trained in simulation, for flinging [9] and folding cloth [33], and for ballistically casting ropes [18].

Finally, environmental modelling has a long history in nonlinear optimal control techniques [28]. From this perspective, it may be difficult to draw an exact line between a simulator and a dynamics model, but the capabilities of control and dynamic planning techniques are steadily encompassing an increasingly rich set of phenomena [29, 30]. Recent research has focused on the differentiability of entire simulators, encompassing various modeling methods and phenomena, from contact-rich rigid body dynamics [7], to mesh-free multiphysics with the Material Point Method [14], to Finite Element based soft-body cutting simulation [12].

B. Closing the Sim-to-Real Gap

In the use of a simulator for any of these purposes, a major concern is the accuracy of approximation of the real world. This “sim-to-real” gap introduces particular concern for RL and optimal control contexts – both seek optimal actions, but an optimal action in an approximate simulator may be quite different from an optimal action on a real-world system. Broadly, there are two ways to solve this problem: by identifying parameters for a given model which more optimally explain the real world, or by choosing a more nuanced or more structurally appropriate model.

In the extreme end of the spectrum of nuanced modelling, deep neural networks or dynamic mode decompositions [16] can be used to directly approximate system input-output dynamics from data [25]. These fully data-driven schemes are able to model highly complex dynamics, for which there may not be expressible equations of motion. However, these schemes can only effectively predict from observations similar to data seen during fitting, and thus they require many example trajectories to generalize even to a single system, much less to similar systems with different physical parameters. Furthermore, despite recent advancements [27, 21], it is difficult to detect when such systems are not making accurate predictions, leading to unsafe and unpredictable actions.

1) *Deformable Object Modeling*: System-specific models require less data and may more easily generalize to similar systems, at the cost of significantly increased human effort required to design the model. Development of nuanced and accurate physical models has been a cornerstone of engineering since

the development of calculus. Civil, mechanical, aerospace [19] and agricultural [26] engineering have long relied on complex models based on solid mechanics and fluid dynamics.

For many applications, computer graphics also requires deformable object simulation. Computer graphics methods are of particular interest to robotics applications due to the shared interest in a tradeoff between fidelity and speed. In computer graphics, it may be desirable to render a reduced model to preserve realtime interactivity, while in robotics, a less accurate dynamics estimate may be much more useful than a high fidelity estimate if it can be provided at a much higher frequency. This tradeoff has been explored in depth for physically-based model reductions of St. Venant-Kirchoff materials using Finite Element Methods [3]. As will be discussed, the approach proposed in this work relies on a constraint formulation to model the grasping hand of the robotic manipulator. Detailed analysis of constrained deformable dynamics, and approaches to overcome related issues of coupling between mesh vertices, is presented in [32].

In robotics, many systems have approximated environments by assuming all objects are rigid. Only recently have common simulators begun to incorporate objects with deformable dynamics, using linked rigid body approximations [30], FEM based dynamics [6], and Position-Based Dynamics (PBD) [23].

2) *Parameter Estimation*: Physically based models often have many physically interpretable parameters. Choosing these parameters poorly can significantly degrade a model’s ability to accurately predict dynamics. *Parameter estimation* is the process of attempting to accurately identify the parameters which most optimally match the system’s true dynamics, and has been an area of interest since the earliest days of robotics, with publications resulting from parameter identifications of individual serial manipulators [2].

Identification of the parameters of a deformable system is also a well explored area. Mechanically focused fields, for which an engineered material will be produced repeatedly within a specified tolerance, are able to sample small sections of material and subject them to detailed stress-strain measurements which specialized scientific apparatus. A survey of such approaches discusses the tradeoffs of relative solution methods with such fine-grained data [20]. In the field of computer graphics, it is valuable to ascertain material properties to get a visually plausible match, so that an animated object with simulated dynamics will look real to a user or audience. An approach for a co-rotational linearly elastic material model is detailed in [31]. In a robotics context, an approach for identifying the parameters of arbitrary material models is given in [11], and is based on the adjoint sensitivity analysis method. In contrast, our proposed method is based on a direct approach, optimizing the parameters jointly with the candidate state trajectories.

III. METHODS

A. Modeling

The true dynamics of a deformable object are continuous in space and time, with an infinite-dimensional state and highly

complex equations of motion. Nevertheless, the mechanics of deformation have a rich history of study dating back to Euler and Lagrange, yielding approximate continuous dynamics in the form of systems of Partial Differential Equations (PDEs). Tractable approximate representations of deformable object mechanics are a cornerstone of engineering analysis for nearly all mechanical systems built in the real world. A classic technique is the Finite Element Method (FEM), which converts the continuous-state system into a discrete-state state Ordinary Differential Equation (ODE). For brevity, we assume familiarity with the FEM technique, and refer interested readers to.

The natural world simultaneously exhibits a complex collection of phenomena, and there are accordingly many aspects of any physical object to model. We focus in this paper on the restricted class of *viscoelastic* deformable objects. Broadly, these objects are governed by restitutive forces that resist local deformations and viscous damping forces that resist movement.

Formally, given a mesh of linear tetrahedral elements with n nodes, and a state vector $\mathbf{x} \in \mathbb{R}^{3n}$ describing the three-dimensional position of each node, the FEM approximation of the equations of motion yields the second order system

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{x}} &= \mathbf{f}_e(\mathbf{x}) + \mathbf{f}_d(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{f}_{ext} \\ \mathbf{f}_d(\mathbf{x}, \dot{\mathbf{x}}) &= -\gamma_r \mathbf{K}(\mathbf{x}) \dot{\mathbf{x}} - \gamma_m \mathbf{M} \dot{\mathbf{x}},\end{aligned}$$

where $\mathbf{M} \in \mathbb{R}^{3n \times 3n}$ is the lumped mass matrix, \mathbf{f}_e describes the internal elastic forces, \mathbf{f}_d describes the collected damping forces, and \mathbf{f}_{ext} collects any external forces, such as gravity. The parameter $\gamma_r \in \mathbb{R}_{>0}$ is a coefficient for an element-local damping term called Rayleigh damping, where $\mathbf{K} = \partial \mathbf{f}_e / \partial \mathbf{x}$ is called the stiffness matrix. The parameter $\gamma_m \in \mathbb{R}_{>0}$ is a coefficient for a global viscous damping term called mass-proportional damping.

1) *Constitutive Model*: The choice of the function \mathbf{f}_e determines the elastic behavior of the system, and is equivalent to the choice of a *constitutive model*. The choice of such a model for a given object is a field in itself [13]. In this work, we select the Neo-hookean model described in [4], primarily for the fact that compressing an element to zero volume requires infinite energy. We emphasize that this choice is based on the real-world properties of the particular object used in our experiments, and our method is compatible with all major viscoelastic constitutive models. The Neo-hookean constitutive model is defined by the energy function

$$\psi(\mathbf{C}) = \frac{1}{2} \lambda (\ln J)^2 - \mu \ln J + \frac{1}{2} \mu (\text{tr } \mathbf{C} - 3)$$

where λ and μ are material parameters called the Lamé parameters, and are effectively a nonlinear change of variables from Poisson's ratio and Young's modulus. For a local deformation gradient \mathbf{F} representing the linearized deformation across a single element,

$$\mathbf{C} = \mathbf{F}^T \mathbf{F} \quad \text{and} \quad J = \det(\mathbf{F}).$$

\mathbf{C} is called the right Cauchy-Green deformation tensor. As mentioned, the $\ln J$ terms result in an infinite amount of energy required to compress an element to zero volume, and thus an infinite restitutive force away from such compression.

2) *Boundary Conditions*: Manipulation problems where the robot can be assumed to have a rigid and unbreakable grasp of the object may be modelled via time-dependent Dirichlet boundary conditions. In the description of a PDE problem, boundary conditions represent the behavior of valid solutions at the edges of the domain. In particular, a Dirichlet boundary condition specifies the value of the solution for regions of the boundary. For the solid mechanics problems considered here, this corresponds to specifying the position and velocity of specific nodes on the boundary of the mesh, representing the rigid grasp of the robot manipulator.

By varying the states specified by the boundary condition with time, we represent the motion of the robot, and excite motion in the rest of the mesh. In our problem formulation, we assume that the robot has rigid control over a subset of “grasped” nodes with indices $I_g \subseteq \{1, \dots, n\}$, and call the state of those nodes $\mathbf{x}^g = (\mathbf{x}_i)_{i \in I_g}$. We model the grasp as a time-varying 6-DOF rigid transform $\mathbf{u}_t \in SE(3)$ of these nodes, resulting in the constraints

$$\mathbf{x}_{t,i} = \mathbf{u}_t \bar{\mathbf{x}}_i \quad \forall i \in I_g, \quad \forall t \in \{1, \dots, T\},$$

where $\bar{\mathbf{x}}_i$ is the undeformed position of node i .

3) *Time Integration*: The second-order FEM dynamics (III-A) can be solved by any ODE solver by introducing variables $\mathbf{v} = \dot{\mathbf{x}}$ and lowering to a first order system. Alternatively, there exist tailored ODE solvers for second-order systems with mass and damping matrices. For general forward simulation tasks, we use the second order implicit Newmark integrator [24], which enables stable timestepping for timesteps on the order of 10 ms.

4) *Measurement model*: We assume that in the real system, we have the ability to measure the positions of the subset of mesh vertices with indices $I_o \subseteq \{1, \dots, n\}$. This is denoted by

$$\mathbf{y} = h(\mathbf{x}) = (x_i)_{i \in I_o}.$$

B. Nonlinear Programming for Parameter Estimation

The model in Section III-A describes an entire family of dynamical systems. Collecting free variables from Equations (III-A) and (III-A1), we see that this family of systems is parameterized by a choice of $(\mathbf{M}, \lambda, \mu, \gamma_r, \gamma_m)$. In this work we assume that the mass matrix \mathbf{M} is known. In experiments, we compute \mathbf{M} by weighing the deformable object, assuming uniform density throughout the mesh, and dividing the resulting mass of each tetrahedron equally among its vertices. (This is known as a lumped mass matrix.) We denote the remaining parameters as $\theta = (\lambda, \mu, \gamma_r, \gamma_m)$. The goal of parameter estimation is to pick system parameters θ^* which best describe a set of collected real-world observations.

We collect data from the real system by supplying a predetermined discrete-time input signal and recording the corresponding output signal. To simplify notation, we present our method using only one pair of input/output signals. However, the method is easily extended to a dataset of multiple input/output signal pairs. Let $U = (\mathbf{u}_1, \dots, \mathbf{u}_T) \in SE(3)^T$ denote the input signal. We execute U with zero-order hold using a constant time

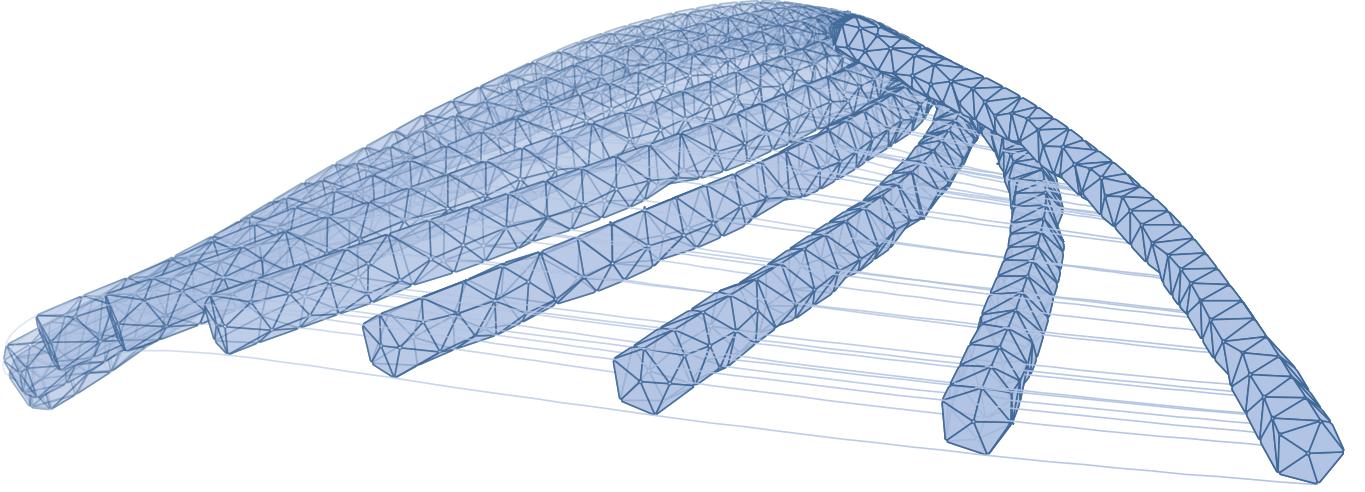


Fig. 2. An example state trajectory

interval of Δt , yielding the output signal $\hat{Y} = (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_T)$. We will also use the notation $X = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ to denote a sequence of mesh states.

Broadly, parameter estimation can be described as the following optimization problem:

$$\begin{aligned} \theta^*, X^* = & \underset{X, \theta}{\operatorname{argmin}} \quad \mathcal{L}(X, \theta) \triangleq \sum_{t=1}^T \|\hat{\mathbf{y}}_t - h(\mathbf{x}_t)\|_2^2 \\ \text{subject to } & X \text{ consistent with} \\ & \text{eqs. (III-A) and (III-A2).} \end{aligned}$$

The choice of squared Euclidean norm to quantify error is arbitrary, and could be replaced by other divergences or metrics.

1) *Equality Constraints:* Given a candidate state trajectory $X = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ and candidate parameters θ , we wish to evaluate how well they satisfy Equation (III-A). At each timestep $t \in \{1, \dots, T\}$, we introduce an equality constraint function

$$\mathbf{c}_t(X, \theta) \triangleq \mathbf{f}_e(\mathbf{x}_t; \theta) + \mathbf{f}_d(\mathbf{x}_t, \mathbf{v}_t; \theta) + \mathbf{f}_{ext} - \mathbf{M}(\theta) \mathbf{a}_t = \mathbf{0},$$

where

$$\mathbf{v}_t = \frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{\Delta t} \quad \text{and} \quad \mathbf{a}_t = \frac{\mathbf{x}_t - 2\mathbf{x}_{t-1} + \mathbf{x}_{t-2}}{\Delta t^2}.$$

In our experiments the system begins in its rest state, so it is reasonable to set $\mathbf{v}_1 = \mathbf{0}$ and for $t \leq 2$, $\mathbf{a}_t = \mathbf{0}$.

Equation (III-B1) applies only for nodes which are not grasped by the manipulator. For the grasped nodes \mathbf{x}^g we introduce a separate equality constraint which enforces the Dirichlet constraint in Equation (III-A2), and supersedes Equation (III-B1):

$$\mathbf{c}_t^g(X, \theta) \triangleq \mathbf{x}_{t,i} - \mathbf{u}_t \bar{\mathbf{x}}_i.$$

In realistic FEM problems, a mesh may have thousands of nodes, and a trajectory may have thousands of timesteps. Many powerful local optimization algorithms rely on the Jacobian of the constraint vector to choose a search direction

for improvement. As such, evaluation of the combined equality constraint \mathbf{c} and its Jacobian require special care for efficient computation. Here we state the equations for $\partial \mathbf{c}_t / \partial X$ and $\partial \mathbf{c}_t / \partial \theta$, with some comments on performant implementation.

$$\begin{aligned} \frac{\partial \mathbf{c}_t}{\partial \mathbf{x}_t} &= \left(1 + \frac{\gamma_r}{\Delta t}\right) \mathbf{K}(\mathbf{x}_t) + \gamma_r \mathbf{H}(\mathbf{x}_t) \cdot \mathbf{v}_t + \frac{\gamma_m}{\Delta t} \mathbf{M} - \frac{1}{\Delta t^2} \mathbf{M} \\ \frac{\partial \mathbf{c}_t}{\partial \mathbf{x}_{t-1}} &= -\frac{1}{\Delta t} (\gamma_r \mathbf{K}(\mathbf{x}_t) + \gamma_m \mathbf{M}) + \frac{2}{\Delta t^2} \mathbf{M} \\ \frac{\partial \mathbf{c}_t}{\partial \mathbf{x}_{t-2}} &= -\frac{1}{\Delta t^2} \mathbf{M} \end{aligned}$$

where the order-3 tensor $\mathbf{H}(\mathbf{x}_t)$, called the Hessian stiffness tensor, has elements $\mathbf{H}_{ijk} = \partial \mathbf{K}_{ij} / \partial \mathbf{x}_{tk}$, and $\mathbf{A} \cdot \mathbf{b} = \mathbf{A}_{ijk} \mathbf{b}^k$ is a tensor contraction along the last dimension. In practice, this term is computed efficiently as the directional derivative $D_{\mathbf{v}} \mathbf{K}(\mathbf{x})$.

As shown in Figure 3, notice that the structure of $\partial \mathbf{c} / \partial \mathbf{x}$ is block sparse, and block lower triangular, and that each block may be computed independently and in parallel. Additionally, each block is itself a sparse matrix arising from the structure of the FEM mesh. Storing $\partial \mathbf{c} / \partial \mathbf{x}$ using a sparse matrix representation is crucial, as the uncompressed matrices can approach terabytes of storage of almost all exact zeros.

2) *Penalty Methods:* Equation (III-B) is difficult to optimize. While the objective is convex, there are a large number of nonlinear equality constraints, so the feasible set is highly nonconvex. Furthermore, the problem is very high dimensional. Satisfying hundreds of simultaneous nonlinear equality constraints is challenging, and we find that many methods reach very suboptimal local minima quite quickly. Relaxing the constraints into a penalty allows optimization algorithms to explore outside of the strictly physically feasible set of trajectories by transforming the constrained Equation (III-B) into the unconstrained problem

$$\underset{X, \theta}{\operatorname{argmin}} \quad \mathcal{L}(X, \theta) + \frac{\mu}{2} \sum_t \|\mathbf{c}_t(X, \theta)\|_2^2,$$

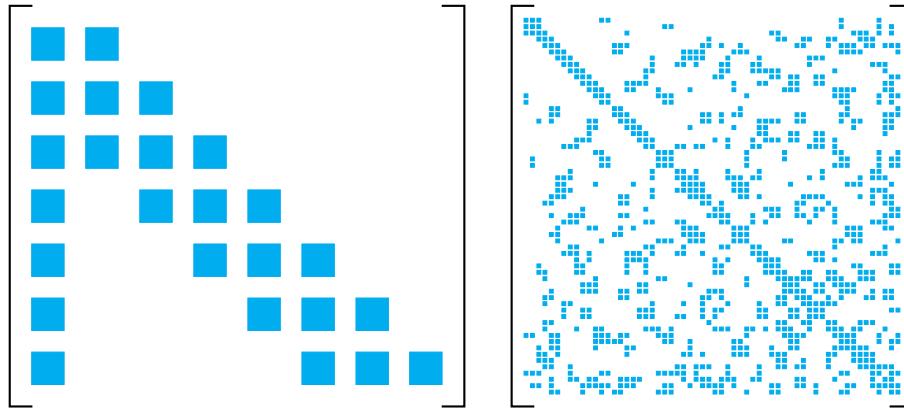


Fig. 3. Structure of the matrices in the computation of $\partial\mathbf{c}/\partial\mathbf{x}$. *Left:* The overall block-diagonal structure of $\partial\mathbf{c}/\partial\mathbf{x}$. The first block column is the parameter Jacobians $\partial\mathbf{c}_t/\partial\theta$. The rest of the matrix is a block structure with a diagonal and two subdiagonals, coming from the time differencing in the calculation of \mathbf{v}_t and \mathbf{a}_t . *Right:* An example symmetric sparsity pattern of one of the diagonal subblocks. This sparsity pattern comes from the connectivity of the nodes in the FEM mesh.

where $\mu > 0$, which we solve using a nonlinear conjugate gradient method [10].

3) μ -Scheduling: Equation (III-B2) leaves a free parameter μ which weights the relative contribution of the equality constraints \mathbf{c} to the output matching loss function \mathcal{L} . As is common in the optimization literature, we find that changing the value of μ during optimization can improve optimizer performance.

4) Initial X Guess: Additionally, the initial guess of the state trajectory X strongly influences optimization convergence. Given an initial guess θ_0 for the parameters, we compute an initial state trajectory as a numerical solution of the dynamics (III-A) where the input \mathbf{u} is determined by the zero-order hold of the sequence U and the initial state \mathbf{x}_1 is set to the solution of the rest-state problem for θ_0 , as detailed below. Since X_0 satisfies Equation (III-A) for θ_0 , it also satisfies $\mathbf{c}_t(\mathbf{x}, \theta_0) = \mathbf{0}$ for all timesteps t , up to small differences caused by the ODE integration scheme. Therefore, it is an approximately feasible point for the optimization problem (III-B).

5) Rest state problem: The rest state problem refers to finding an equilibrium point of the FEM dynamics under the Dirichlet boundary condition, that is, solving the nonlinear system of equations

$$\mathbf{f}_e(\mathbf{x}) = -\mathbf{f}_{ext}, \quad \mathbf{x}_i = \mathbf{u}\bar{\mathbf{x}}_i \quad \forall i \in I_g$$

for \mathbf{x} with \mathbf{u} and θ fixed. When solving this system numerically, we must take precautions to ensure that the solver does not query FD for a state \mathbf{x} with very large restorative forces caused by the $\ln \det(\mathbf{F})$ term in the Neo-Hookean constitutive model (III-A1). In our experiments it was sufficient to use Newton's method with a backtracking line search to ensure that the stiffness matrix \mathbf{K} remains positive definite.

IV. PRELIMINARY RESULTS

We implement a FEM simulation environment in the Julia language [5], and use the Optim [22] software suite's implementation of a nonlinear conjugate gradient algorithm to optimize

Parameter	Ground truth	Estimated
λ	200.0	734.3
μ	300.0	182.8
γ_r	0.009	0.021
γ_m	1.0	0.91

TABLE I
COMPARISON OF GROUND-TRUTH AND ESTIMATED VALUES FOR THE MATERIAL PARAMETERS θ IN OUR EXPERIMENT. (λ, μ): LAMÉ PARAMETERS. (γ_r, γ_m): RAYLEIGH DAMPING AND MASS-PROPORTIONAL DAMPING CONSTANTS.

Equation (III-B2). To store the sparse Jacobian $\partial\mathbf{c}/\partial\mathbf{x}$, we use the common Compressed Sparse Column (CSC) format. We are able to evaluate the Jacobian of a trajectory of 100 timesteps, with a mesh of 176 3-dimensional nodes in less than 100 ms (minimum 57 ms, median 82 ms, mean 98 ms).

Given an initial parameter guess θ_0 , we are able to identify a set of parameters and state variable which achieve a mean optimization loss $\mathcal{L}(X, \theta)$ of 1×10^{-3} m mean Euclidean error. However, it is important to note that due to the penalty method presented, there is still slack in the dynamics constraints during optimization, and that these parameters do not generate similar rollouts during forward simulation, where an individual point on the distal tip of the mesh has a mean 15 cm tracking error. This points towards possible issues with our choice of a penalty-based optimization algorithm or towards a difference between the implicit Newmark integrator and the finite-difference approximations enforced in the dynamics constraints $\mathbf{c}_t(X, \theta)$. This provides motivation for ongoing work.

V. CONCLUSION AND ONGOING WORK

We have proposed a method to estimate the material parameters of a deformable object by observing the object's motion under non-destructive robotic manipulation. Our method is intended to improve the fidelity of finite-element models in simulation environments for complex manipulation tasks.

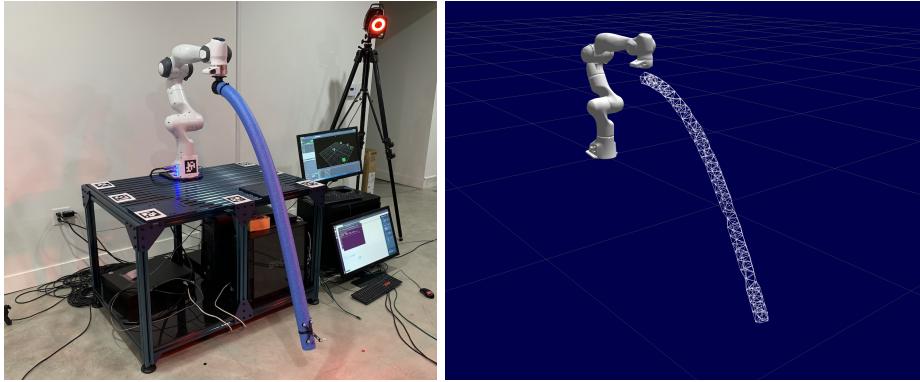


Fig. 4. A motivating testbed problem setup for our proposed method. *Left*: Franka Emika Panda arm rigidly grasping a foam cylinder, with attached Vicon markers. *Right*: Visualization of a simulated FEM mesh used for the methods described in this paper.

Unlike approaches based on the adjoint method or shooting, our method is similar to collocation-style trajectory optimization and estimates the unobserved trajectory of the finite-element mesh along with the material parameters. We believe that this method will show an improved set of initial guesses which will converge to a correct system parameter identification, compared to other methods in the literature.

Our ongoing work focuses on three main efforts. First, we intend to compare the accuracy and computational cost of a wider range of constrained nonlinear optimization algorithms for the collocation-style problem setup. Second, we will investigate the possibility of exploiting the problem structure for faster or more accurate results, such as solving a series of estimation problems over increasing time horizons. Third, we will compare our method against adjoint- and shooting-based parameter estimation algorithms. We are especially interested in comparing stability-related properties such as sensitivity to initial guess.

REFERENCES

- [1] Ilge Akkaya et al. “Solving Rubik’s Cube with a Robot Hand”. 2019. arXiv: 1910.07113.
- [2] Brian Armstrong, Oussama Khatib, and Joel Burdick. “The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm”. In: *Proceedings. 1986 IEEE International Conference on Robotics and Automation*. Vol. 3. IEEE. 1986, pp. 510–518.
- [3] Jernej Barbič and Doug L James. “Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models”. In: *ACM transactions on graphics (TOG)* 24.3 (2005), pp. 982–990.
- [4] Ted Belytschko et al. *Nonlinear Finite Elements for Continua and Structures*. John wiley & sons, 2014.
- [5] Jeff Bezanson et al. “Julia: A Fresh Approach to Numerical Computing”. In: *SIAM Review* 59.1 (2017), pp. 65–98. DOI: 10.1137/141000671.
- [6] Erwin Coumans and Yunfei Bai. “PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning”. In: (2016). URL: <http://pybullet.org>.
- [7] Filipe de Avila Belbute-Peres et al. “End-to-End Differentiable Physics for Learning and Control”. In: *Advances in neural information processing systems* 31 (2018).
- [8] Hilding Elmquist, Sven Erik Mattsson, and Martin Otter. “Modelica-a Language for Physical System Modeling, Visualization and Interaction”. In: *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (Cat. No. 99TH8404)*. IEEE. 1999, pp. 630–639.
- [9] Huy Ha and Shuran Song. “Flingbot: The Unreasonable Effectiveness of Dynamic Manipulation for Cloth Unfolding”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 24–33.
- [10] William W. Hager and Hongchao Zhang. “Algorithm 851: CGDESCENT, a Conjugate Gradient Method with Guaranteed Descent”. In: *ACM Trans. Math. Softw.* 32.1 (Mar. 2006), pp. 113–137. ISSN: 0098-3500. DOI: 10.1145/1132973.1132979. URL: <https://doi.org/10.1145/1132973.1132979>.
- [11] David Hahn et al. “Real2sim: Visco-elastic Parameter Estimation from Dynamic Motion”. In: *ACM Transactions on Graphics (TOG)* 38.6 (2019), pp. 1–13.
- [12] Eric Heiden et al. “Disect: A Differentiable Simulation Engine for Autonomous Robotic Cutting”. In: *Robotics: Science and Systems* (2021).
- [13] Gerhard .A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley, 2000. ISBN: 978-0-471-82304-9.
- [14] Yuanming Hu et al. “DiffTaichi: Differentiable Programming for Physical Simulation”. In: *ICLR* (2020).
- [15] Nathan Koenig and Andrew Howard. “Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator”. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. Sept. 2004, 2149–2154 vol.3. DOI: 10.1109/IROS.2004.1389727.
- [16] J Nathan Kutz et al. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.

- [17] Sergey Levine et al. “Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection”. Aug. 28, 2016. doi: 10.48550/arXiv.1603.02199. arXiv: 1603.02199 [cs]. URL: <http://arxiv.org/abs/1603.02199>.
- [18] Vincent Lim et al. “Planar Robot Casting with Real2Sim2Real Self-Supervised Learning”. 2021. arXiv: 2111.04814.
- [19] Richard H. MacNeal and Caleb W. McCormick. “The NASTRAN Computer Program for Structural Analysis”. In: *Computers & Structures* 1.3 (1971), pp. 389–412.
- [20] Rolf Mahnken. “Identification of Material Parameters for Constitutive Equations”. In: *Encyclopedia of Computational Mechanics Second Edition*. John Wiley & Sons, Ltd, 2017, pp. 1–21. ISBN: 978-1-119-17681-7. doi: 10.1002/9781119176817.ecm2043. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119176817.ecm2043>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119176817.ecm2043>.
- [21] Peter Mitrano, Dale McConachie, and Dmitry Berenson. “Learning Where to Trust Unreliable Models in an Unstructured World for Deformable Object Manipulation”. In: *Science Robotics* 6.54 (2021), eabd8170. doi: 10.1126/scirobotics.abd8170. eprint: <https://www.science.org/doi/pdf/10.1126/scirobotics.abd8170>. URL: <https://www.science.org/doi/abs/10.1126/scirobotics.abd8170>.
- [22] Patrick Kofod Mogensen and Asbjørn Nilsen Riseth. “Optim: A Mathematical Optimization Package for Julia”. In: *Journal of Open Source Software* 3.24 (2018), p. 615. doi: 10.21105/joss.00615.
- [23] Matthias Müller et al. “Detailed Rigid Body Simulation with Extended Position Based Dynamics”. In: *Computer Graphics Forum*. Vol. 39. 8. Wiley Online Library. 2020, pp. 101–112.
- [24] Nathan M. Newmark. *A Method of Computation for Structural Dynamics*. American Society of Civil Engineers, 1959.
- [25] James A. Preiss et al. “Tracking Fast Trajectories with a Deformable Object Using a Learned Model”. In: IEEE Conference on Robotics and Automation 2022. 2022.
- [26] Alan R. Reece. “The Fundamental Equation of Earth-Moving Mechanics”. In: *Proceedings of the Institution of Mechanical Engineers, Conference Proceedings*. Vol. 179. 6. SAGE Publications Sage UK: London, England. 1964, pp. 16–22.
- [27] Apoorva Sharma, Navid Azizan, and Marco Pavone. “Sketching Curvature for Efficient Out-of-Distribution Detection for Deep Neural Networks”. In: *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. Ed. by Cassio de Campos and Marloes H. Maathuis. Vol. 161. Proceedings of Machine Learning Research. PMLR, July 27–30, 2021, pp. 1958–1967. URL: <https://proceedings.mlr.press/v161/sharma21a.html>.
- [28] Jean-Jacques E Slotine, Weiping Li, et al. *Applied Nonlinear Control*. Vol. 199. 1. Prentice hall Englewood Cliffs, NJ, 1991.
- [29] Yuval Tassa, Tom Erez, and Emanuel Todorov. “Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 4906–4913.
- [30] Emanuel Todorov, Tom Erez, and Yuval Tassa. “Mujoco: A Physics Engine for Model-Based Control”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.
- [31] Bin Wang et al. “Deformation Capture and Modeling of Soft Objects.” In: *ACM Trans. Graph.* 34.4 (2015), pp. 94–1.
- [32] Bohan Wang, Mianlun Zheng, and Jernej Barbič. “Adjustable Constrained Soft-Tissue Dynamics”. In: *Computer Graphics Forum* 39.7 (2020), pp. 69–79. ISSN: 1467-8659. doi: 10.1111/cgf.14127. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14127>.
- [33] Thomas Weng et al. “FabricFlowNet: Bimanual Cloth Manipulation with a Flow-Based Policy”. In: *Conference on Robot Learning*. 2021.