2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Dynamic Robotic Cloth Folding with Efficient Koopman Operator-Based Model Predictive Control

Edoardo Caldarelli[1], Franco Coltraro[2], Adrià Colomé[2], Lorenzo Rosasco[1, 3], and Carme Torras[2]

***Abstract*— Robotic cloth folding is a challenging task, particularly when considering dynamic folding tasks, which aim at folding cloth by fast motions that leverage its dynamics. When subject to such fast motions, the complexity of cloth dynamics hinders both system identification and planning of folding trajectories, resulting in a difficult simulation-to-reality transfer when using physical models of cloth. Compared to the dexterity that humans exhibit when performing folding tasks, robotic approaches usually employ small garments with quite rigid dynamics, and are either too slow, or fast but imprecise, requiring several attempts to achieve a reasonably good fold. In this paper, we tackle these challenges by generating fast folding trajectories with a novel model predictive controller, integrating physics-based simulation of cloth dynamics and efficient, kernel-based Koopman operator regression. Koopman operator regression, an increasingly popular machine learning technique for nonlinear system identification, is used to obtain a linear model for the cloth being folded. Such a surrogate model, trained with data from a high-fidelity, physics-based cloth simulator, can then be employed within a suitable model predictive control algorithm, in place of the costly, nonlinear one, to efficiently generate folding trajectories to be executed by a robotic manipulator. Both in simulated and real-robot experiments, we show how the linearization supplied by the Koopman operator-based model can be employed to efficiently generate fast folding trajectories to unseen poses, without sacrificing folding accuracy.**

I. Introduction and Related Works

Robotic cloth manipulation is an increasingly relevant area of research, which greatly challenges classic robot control algorithms [1], [2]. The deformable nature of cloth requires state-of-the-art techniques in order to simulate [3], [4] and control [5], [6], [7], [8] the cloth dynamics within a robotic platform, with the ultimate goal of autonomously performing cloth manipulation skills with the same degree of expertise humans exhibit in their everyday lives, as shown, e.g., in Fig. 1.

Within the context of robotic cloth manipulation, we can distinguish between *quasi-static* tasks, and *dynamic* manipulation skills [9]. In this paper, we focus on the latter case: We propose an algorithm for *cloth folding* that leverages the dynamics of the cloth, to completely fold a rectangular cloth by acting only on one of its corners. Namely, we aim at performing such a task with a single manipulator and a single robot motion [7], [8], therefore considering a more resource-efficient environment compared to bimanual setups, or single-handed, quasi-static ones, which typically require multiple motion primitives [10]. Besides domestic and assistive environments, dynamic manipulation is of particular importance in industrial settings where robots could automatize the manual folding of cloth. Speed is critical for efficiency as evidenced by the fact that trained humans that work in textile factories always fold clothes dynamically and in a very fast fashion.

Fig. 1: An example of a dynamic cloth folding trajectory performed by a human expert. Albeit efficient, the high speed of the motion may hinder the accuracy of the fold, if the cloth dynamics are not taken into consideration.

In our setting, dynamics-aware skills become of paramount importance, due to the high inertia of the cloth when manipulated fast. These inertial phenomena are known to cause severe sim-to-real gaps [11]. Nonetheless, as shown in our experiments, we are able to achieve a successful zero-shot sim-to-real transfer on fast folding actions (less than 1.5 s), by deploying the control strategy described below. The dynamic folding skills are achieved by means of a novel model predictive control (MPC) framework, which integrates both an accurate, and a data-driven efficient model of the cloth dynamics. Specifically, high-fidelity folding data are generated with a physics-based simulator [4], and use to train a non-parametric machine learning approximation, obtained through *Koopman operator regression* [13], [14], [15], [12]. Koopman operator regression recasts the nonlinear dynamics of cloth to linear ones (albeit in an infinite dimensional vector space), that can be learned efficiently [12] and embedded in a *linear* MPC pipeline. MPC can then be used to generate trajectories that allow the robot to fold the cloth to unseen target poses. Most importantly, those trajectories account for *performance criteria* and *user-defined* constraints (as opposed to the training data), which can be easily embedded in the optimal control problem underlying MPC.

In our control architecture, the high fidelity simulator from

[1]Istituto Italiano di Tecnologia, Genoa, Italy [2]Institut de Robòtica i Informàtica Industrial, CSIC–UPC, Barcelona, Spain [3]MaLGa Center, DIBRIS, Università degli Studi di Genova, Genova, Italy. Correspondence to: edoardo.caldarelli@iit.it.

This work was partially funded by the European Project HORIZON-CL4-2024-DIGITAL-EMERGING-01-101189600 (FlexCycle). E. Caldarelli acknowledges support from COST Action InterCoML (CA24136). F. Coltraro was fully supported by Momentum CSIC Programme project MMT24-IRII-01 and now is partially by ClothIRI (CSIC 202350E080) project. L. Rosasco acknowledges the financial support of the European Commission (Horizon Europe grant ELIAS 101120237), and the Ministry of Education, University and Research (FARE grant ML4IP R205T7J2KP).

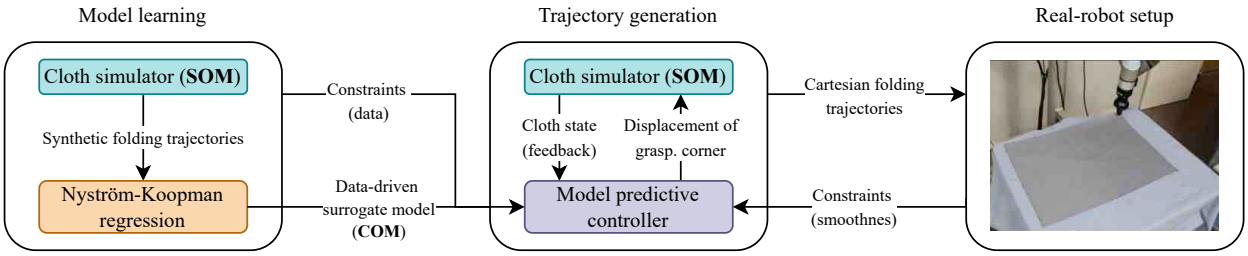


Fig. 2: Summary of the cloth folding strategy proposed in this paper. A cloth simulator [4] simulates synthetic folding trajectories, which are used as training data for a Koopman operator regression model [12]. The data-driven model, along with suitable constraints, are fed to a model predictive controller. In this way, a folding trajectory to a given target pose is generated, accounting for performance requirements (speed and accuracy of the fold) as well as constraints stemming from the real robot setup (e.g., smoothness of the velocities). The trajectory is then executed on the real robot.

[4] acts as a *simulation-oriented model* (SOM), whereas the Koopman approximation from [12] serves as a *control-oriented model* (COM). The SOM can be used to provide accurate feedback in simulation, whereas the COM can be used to perform cheap, but reliable forecasts of the cloth dynamics without needing any control or position derivatives of the SOM. Solving the MPC with these two models yields a folding trajectory that can then be executed with a real robot, to accomplish the folding task. The integration of the components yielding our cloth folding strategy is detailed in Fig. 2.

Contributions: In summary, in this paper we:

- combine the physics-based, collision-aware SOM from [4] with the efficient Koopman operator regression algorithm studied in [12], obtaining a data-driven, linear COM for a piece of cloth;
- use such a COM in a linear MPC control strategy, to generate constrained robot trajectories, in closed-loop with the SOM, folding a piece of cloth to an unseen target pose;
- execute and evaluate the generated trajectories on a real robotic platform, showcasing how our methodology reduces the sim-to-real gap. Most importantly, the folding motions happen in less than 1.5 s, counteracting the inertial effects of cloth due to such a high speed.

Outline: The structure of this paper is as follows: Section II describes the setup we consider in this work, reporting the details of the SOM for the cloth. Section III details the COM, and the data-driven MPC controller we propose in this paper. Section IV assesses the proposed control pipeline in simulated and real robot experiments. Lastly, Section V contains the conclusions.

II. Physical Model of the Cloth

In this section we give a self-contained presentation of the inextensible cloth model developed in [3], [4] so that it can be used to generate the data needed to train the Koopman operator regressor, which will be in turn used as the backbone of a novel MPC strategy for dynamic cloth folding.

We assume that the piece of cloth S we wish to control has been discretized into a quadrilateral mesh and the position of all its N vertices or nodes (denoted by $p_i(t) = [x_i(t), y_i(t), z_i(t)]^\intercal \in \mathbb{R}^3$) at time $t \geq 0$ is given by $\boldsymbol{\varphi}(t) = (\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t))^\intercal \in \mathbb{R}^{3N}$, where $\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)$ denote the x-coordinates (resp. y and z) of all the nodes of the discrete surface in $\mathbb{R}^N$ at time t and $\mathbf{v}^\intercal$ denotes the tranpose of $\mathbf{v}$.

Then, the inextensible cloth model including aerodynamics but without collisions consists in the following set of ordinary differential equations (ODEs):

$$\begin{cases} \rho \mathbf{M}\ddot{\boldsymbol{\varphi}} = -\delta \mathbf{M}\mathbf{g} - \kappa \mathbf{K}\boldsymbol{\varphi} - \alpha \mathbf{M}\dot{\boldsymbol{\varphi}} - \nabla \mathbf{C}(\boldsymbol{\varphi})^\intercal \boldsymbol{\lambda} \\ \mathbf{C}(\boldsymbol{\varphi}) = 0, \end{cases} \tag{1}$$

where the meaning of each term and parameter (see also Table I) is:

- $\rho > 0$ is the density of the cloth (assumed to be homogeneous, see Table II) and $\mathbf{M}$ is the (diagonal) mass matrix where each $m_k > 0$ is one fourth of the sum of the areas of all incident quads to the node p_k,
- the term $-\delta \mathbf{M}\mathbf{g}$ accounts for the force of gravity, where $\mathbf{g} = [0, \dots, 0 | 0, \dots, 0 | g, \dots, g]^\intercal$ and $g = 9.8\text{m/s}^2$. In order to model the aerodynamic effects of air resistance on cloth we allow the inertial ρ and gravitational δ masses to be different (for a detailed discussion and justification of this simplified aerodynamics model, see [16]) introducing thus an upwards constant lift force. Hence, we set $\delta \leq \rho$ as a new parameter of the model which we call the *virtual mass*,
- the stiffness matrix (we are using the isometric bending model described in [17]) is $\mathbf{K} = \mathbf{L}^\intercal \mathbf{M} \mathbf{L}$ where $\mathbf{L}$ is an approximation of the point-wise Laplacian and $\kappa > 0$ is a bending constant,
- the term $\alpha \mathbf{M}\dot{\boldsymbol{\varphi}}$ is called Rayleigh damping and the magnitude of $\alpha > 0$ is responsible for modeling the dampening of slow oscillations of the cloth and the drag of air [18],
- and finally $\boldsymbol{\lambda}(t)$ are the Lagrange multipliers ensuring *inextensibility* (to be described next) and other possible positional constraints, in our case manipulating the textile by prescribing the position of one of its corners.

Besides the aforementioned positional constraints introduced by manipulation, the smooth function $\mathbf{C} : \mathbb{R}^{3N} \to \mathbb{R}^{n_C}$ is responsible for modeling the *inextensibility* of the

Parameter	Meaning
ρ	Density (inertial mass)
δ	Virtual (gravitational) mass
κ	Bending/stiffness
α	Damping of slow oscillations

TABLE I: Physical parameters of the inextensible cloth model relevant for the folding task considered and their meaning.

textile through the satisfaction of the equality constraints $\mathbf{C}(\boldsymbol{\varphi}) = \mathbf{0}$. Each constraint $C_i(\boldsymbol{\varphi}(t)) = 0,\ i = 1, \dots, n_C$ is in fact a quadratic function of its argument and we have n_C of them, depending on the number of nodes N of the discretization (for more details, see [3]).

The inextensibility constraints $\mathbf{C}(\boldsymbol{\varphi}) = \mathbf{0}$ model what is usually called the *internal dynamics* of cloth; furthermore, for its application in our scenario, we also need to include collisions of the cloth with the table and with itself. We model this by enforcing a set $H_i(\boldsymbol{\varphi}(t)) \geq 0$, $i = 1, \dots, n_H$ of n_H inequality constraints $\mathbf{H}(\boldsymbol{\varphi}) \geq \mathbf{0}$, such that the differentiable function $\mathbf{H} : \mathbb{R}^{3N} \to \mathbb{R}^{n_H}$ satisfies that for each $i = 1, \dots, n_H$ we have a non-zero outwards normal $\nabla H_i(\boldsymbol{\varphi}) \neq \mathbf{0}$ (e.g. for a table locally $\nabla H_i(p_k) = (0, 0, 1)$). We can then model (self)collisions together with friction by including non-smooth forces into the equations of motion of the cloth (this is known as Signorini's contact model, see [4] for more details).

A. Numerical Integration of the System

To integrate the system numerically from time t_j to $t_{j+1} = t_j + \Delta t$ given positions $\boldsymbol{\varphi}_j$ and velocities $\dot{\boldsymbol{\varphi}}_j$ at t_j, we perform an iterative process

$$\boldsymbol{\varphi}^{i+1} = \boldsymbol{\varphi}^i + \Delta\boldsymbol{\varphi}^{i+1} \tag{2}$$

where the initial point is the unconstrained step $\boldsymbol{\varphi}^0$ given by applying an implicit Euler scheme to the equations of motion (1) ignoring all equality and inequality constraints. Moreover, we write:

$$\mathbf{H}(\boldsymbol{\varphi}^{i+1}) = \mathbf{H}(\boldsymbol{\varphi}^i + \Delta\boldsymbol{\varphi}^{i+1}) \simeq \mathbf{H}(\boldsymbol{\varphi}^i) + \nabla\mathbf{H}(\boldsymbol{\varphi}^i)\Delta\boldsymbol{\varphi}^{i+1},$$

and similarly

$$\mathbf{C}(\boldsymbol{\varphi}^{i+1}) = \mathbf{C}(\boldsymbol{\varphi}^i + \Delta\boldsymbol{\varphi}^{i+1}) \simeq \mathbf{C}(\boldsymbol{\varphi}^i) + \nabla\mathbf{C}(\boldsymbol{\varphi}^i)\Delta\boldsymbol{\varphi}^{i+1},$$

and then solve iteratively for $\mathbf{q} := \Delta\boldsymbol{\varphi}^{i+1}$ the following sequence of quadratic programs with linear equality and inequality constraints:

$$\begin{cases} \min_{\mathbf{q}} \frac{1}{2}\mathbf{q}^\intercal \cdot \mathbf{M} \cdot \mathbf{q} - \mathbf{q}^\intercal \cdot \mathbf{f}_\mu(\dot{\boldsymbol{\varphi}}^i) \\ \mathbf{C}(\boldsymbol{\varphi}^i) + \nabla\mathbf{C}(\boldsymbol{\varphi}^i)\mathbf{q} = \mathbf{0}, \\ \mathbf{H}(\boldsymbol{\varphi}^i) + \nabla\mathbf{H}(\boldsymbol{\varphi}^i)\mathbf{q} \geq \mathbf{0}, \end{cases} \tag{3}$$

where $\dot{\boldsymbol{\varphi}}^i = \frac{\boldsymbol{\varphi}^i - \boldsymbol{\varphi}_j}{\Delta t}$ is the approximation of $\dot{\boldsymbol{\varphi}}_{j+1}$ at iteration i and $\mathbf{f}_\mu(\dot{\boldsymbol{\varphi}}^i)$ accounts for Coloumb's friction. For more theoretical and implementation details, see [4].

B. Parameter Estimation for the Cloth and a Priori Formulas

The previously described physical model shows great accuracy in describing a wide variety of real cloth motions including very dynamic movements and fast collisions within a margin of error of 1 cm (see [3] for an experimental validation with depth cameras and [4] for one with a motion capture system). However, by virtue of introducing the aerodynamics parameter $\delta > 0$ (the virtual mass), the optimal value of the physical parameters of Table I varies not only with each different textile but also with the speed at which it moves (except for the density ρ which is constant for each fabric).

Therefore, in [16] *a priori* formulas for the values of the parameters α and δ where developed:

$$\begin{aligned} \hat{\delta} &= -0.0223 - 0.0178S + 0.0714V + 0.7664\rho, \\ \hat{\alpha} &= +0.2082 - 0.1481S + 1.1804V + 1.7440\rho, \end{aligned} \tag{4}$$

where ρ is the density of the cloth, S is a normalized area measure (in our case it will be always 2) and V is the average (in time and over all nodes of the cloth) of 50% of the highest squared velocities. Notice that this formula allows us to deduce the value of the cloth's parameters depending only on its density, size and speed. The coefficients of the linear formulas (4) were found by using optimization with several real Motion Capture (MoCap) recordings of textiles of different sizes moving at various speeds. In [16] it was shown that using formulas (4) gives a comparable absolute error to using the optimal values of α and δ found by minimizing cloth errors for each individual recording.

Lastly, it is relevant to mention that in the MoCap recordings used for obtaining formulas (4), the stiffness κ of the cloths had almost no influence in the motions and hence could not be estimated reliably. Nevertheless, for the dynamic folding task considered in this paper, this parameter (along with α and δ) has a big influence on the motion of the cloths. Although this parameter κ could be estimated by using a suitable version of a simulated Cusick test, see [3], in this work we empirically tune that parameter, based on the associated performance of the controller.

III. Proposed Controller Design

The controller deployed in this article is an MPC approximation of the infinite-horizon LQR studied in [12].

A. Surrogate System Identification

In this work, we rely on the system identification techniques developed in [12]. There, the nonlinear data, obtained through the system's dynamics (in our case obtained by iteratively solving (3)), are lifted to an infinite-dimensional space, where the dynamics are linear through the *Koopman operator*. Then, this infinite-dimensional space is approximated with a finite-dimensional one, through a data-based approach, i.e., the Nyström method. Specifically, we solve here a regularized least-squares problem, to learn a surrogate dynamical model for the cloth being folded. In this subsection, we give an overview of the steps taken to build such surrogate model, to later use it as a COM in a MPC in Section III-B.

1) Learning problem: When considering the dynamics of the cloth being folded, we classically take as control input the displacement (i.e., the variation in position) of the corner grasped by the robot [3]. However, this modeling choice may be unrealistic for real-robot scenarios, since it requires to have a perfect point grasp of the corner considered. In this work, we relax this assumption by modeling a *vertical grasp* of one corner, with fixed orientation of the robot's end-effector, i.e., we control the position of *two* adjacent points in the cloth mesh. As we will show in this section, imposing such a constant orientation can be naturally encoded as a constraint for the MPC. In the following, the controlled displacement will be denoted as $\Delta\mathbf{u} \in \mathbb{R}^6$. The discretization described in Section (II-A) allows to measure triples of the form $(\boldsymbol{\varphi}_j, \Delta\mathbf{u}_j, \boldsymbol{\varphi}_{j+1})$ where $\boldsymbol{\varphi}_j$ represents the state of the vertices of the cloth mesh at time t_j. For a suitable flow map $f : \mathbb{R}^{3N} \times \mathbb{R}^6 \to \mathbb{R}^{3N}$, we aim at building a data-driven surrogate for the transitions

$$\boldsymbol{\varphi}_{j+1} = f(\boldsymbol{\varphi}_j, \Delta\mathbf{u}_j), \tag{5}$$

for $j \geq 0$. This is done by means of Koopman operator regression with approximated reproducing kernels [12].

2) State transformation and regression problem: Let us consider a reproducing kernel Hilbert space $\mathcal{H}$, associated to the kernel function $k : \mathbb{R}^{3N} \times \mathbb{R}^{3N} \to \mathbb{R}$. Let $\psi : \mathbb{R}^{3N} \to \mathcal{H}$ be the corresponding canonical feature map. After having collected a set of n_{train} training inputs (i.e., a tuple $(\boldsymbol{\varphi}_i, \Delta\mathbf{u}_i)$) and training outputs (i.e., $\boldsymbol{\varphi}_i^+ = f(\boldsymbol{\varphi}_i, \Delta\mathbf{u}_i)$), we set up the following regression problem, to be solved over the set of linear operators from $\mathcal{H} \times \mathbb{R}^6$ to $\mathcal{H}$:

$$G = \arg\min_{\mathbf{W}:\mathcal{H}\times\mathbb{R}^6\to\mathcal{H}} \mathcal{R}(\mathbf{W}) + \gamma \left\|\mathbf{W}\right\|_{\text{HS}}^2, \tag{6}$$

where

$$\mathcal{R}(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n_{\text{train}}} \left\| \psi(\boldsymbol{\varphi}_i^+) - \mathbf{W}\begin{bmatrix} \psi(\boldsymbol{\varphi}_i) \\ \Delta\mathbf{u}_i \end{bmatrix} \right\|_{\mathcal{H}}^2. \tag{7}$$

As shown by [12], (6) can be approximated with the Nyström method, to improve the computational efficiency of the solution [19], [20].

3) Nyström approximation: The Nyström approach entails sub-sampling m_{nys} points from the training set, named *Nyström landmarks* and denoted as $\{\tilde{\boldsymbol{\varphi}}_i\}_{i=\{1\dots m_{\text{nys}}\}}$. Let $\boldsymbol{\Pi} : \mathcal{H} \to \mathcal{H}$ be the orthogonal projector on $\text{span}\{k(\boldsymbol{\varphi}_1, \cdot), \dots, k(\boldsymbol{\varphi}_{m_{\text{nys}}}, \cdot)\}$, and let $\boldsymbol{\Pi}_{\text{in}} : \mathcal{H} \times \mathbb{R}^6 \to \mathcal{H} \times \mathbb{R}^6$ be the block diagonal projector $\boldsymbol{\Pi}_{\text{in}}\begin{bmatrix} \psi(\boldsymbol{\varphi}) \\ \Delta\mathbf{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Pi}\psi(\boldsymbol{\varphi}) \\ \Delta\mathbf{u} \end{bmatrix}$. Problem (6) is replaced by

$$\mathbf{G} = \arg\min_{\mathbf{W}:\mathcal{H}\times\mathbb{R}^6\to\mathcal{H}} \mathcal{R}(\boldsymbol{\Pi}\mathbf{W}\boldsymbol{\Pi}_{\text{in}}) + \gamma \left\|\mathbf{W}\right\|_{\text{HS}}^2. \tag{8}$$

Solving this problem allows to retrieve a linear operator that evolves the dynamics of the transformed state forward in time. Namely, for an initial state $\boldsymbol{\varphi}_0$,

$$\begin{cases} z_0 = \psi(\boldsymbol{\varphi}_0), \\ z_{t+1} = \mathbf{G}\begin{bmatrix} z_t \\ \Delta\mathbf{u}_t \end{bmatrix}. \end{cases} \tag{9}$$

4) Surrogate vector-valued dynamics: As shown in [12], the dynamics in (9) can be equivalently expressed in terms of vectors rather than functions in $\mathcal{H}$. The closed-form expressions are given as follows. Let $\mathcal{B}(\cdot,\cdot)$ be a block-diagonal operator s.t. $\mathcal{B}(\mathbf{M}, \mathbf{N}) = \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{N} \end{bmatrix}$. We can further define $\delta_{i,j} = 1 \iff i = j$, $\alpha > 0$, and the following kernel matrices:

- $\mathbf{k}_{m,\boldsymbol{\varphi}} \in \mathbb{R}^{m_{\text{nys}}}, (\mathbf{k}_{m,\boldsymbol{\varphi}})_i = k(\tilde{\boldsymbol{\varphi}}_i, \boldsymbol{\varphi})$;
- $\mathbf{K}_m \in \mathbb{R}^{m_{\text{nys}}\times m_{\text{nys}}}, (\mathbf{K}_{m,m})_{i,j} = k(\tilde{\boldsymbol{\varphi}}_i, \tilde{\boldsymbol{\varphi}}_j) + \alpha\delta_{i,j}$;
- $\mathbf{S}_{\mathbf{u}} \in \mathbb{R}^{n_{\text{train}}\times 3}$, $(\mathbf{S}_{\mathbf{u}})_i = \frac{1}{\sqrt{n}}\mathbf{u}_i^\intercal$;
- $\mathbf{K}_{nm} \in \mathbb{R}^{n_{\text{train}}\times m_{\text{nys}}}$, $(\mathbf{K}_{nm})_{i,j} = k(\boldsymbol{\varphi}_i, \tilde{\boldsymbol{\varphi}}_j)$
- $\mathbf{K}_{mn} = \mathbf{K}_{nm}^\intercal$.

For initial conditions

$$\mathbf{z}_0 = (\mathbf{K}_m^\dagger)^{1/2}\mathbf{k}_{m,\boldsymbol{\varphi}_0}, \tag{10}$$

we can define the following data-driven system for $t \geq 0$ [12]:

$$\begin{aligned} \mathbf{z}_{t+1} = (\mathbf{K}_m^\dagger)^{1/2}\mathbf{K}_{mn}&\begin{bmatrix} \mathbf{K}_{nm} & \sqrt{n}\mathbf{S}_{\mathbf{u}} \end{bmatrix} \\ &\cdot \left(\begin{bmatrix} \mathbf{K}_{mn} \\ \sqrt{n}\mathbf{S}_{\mathbf{u}}^T \end{bmatrix}\begin{bmatrix} \mathbf{K}_{nm} & \sqrt{n}\mathbf{S}_{\mathbf{u}} \end{bmatrix} + \gamma n\mathcal{B}(\mathbf{K}_m, I) \right)^\dagger \\ &\cdot \mathcal{B}(\mathbf{K}_m^{1/2}, I)\begin{bmatrix} \mathbf{z}_t \\ \mathbf{u}_t \end{bmatrix}. \end{aligned} \tag{11}$$

5) State reconstruction: After defining these data-driven dynamics, we may be interested in reconstructing an approximated value for the state $\boldsymbol{\varphi}$ from the surrogate state $\mathbf{z}$. Following [12], we do so by means of the following reconstruction matrix, obtained after stacking the regression outputs in the matrix $\mathbf{X}^+ \in \mathbb{R}^{3N\times n}$:

$$\mathfrak{C} = \mathbf{X}^+\mathbf{K}_{nm}(\mathbf{K}_{mn}\mathbf{K}_{nm} + \lambda n\mathbf{K}_m)^{-1}\mathbf{K}_m^{1/2}. \tag{12}$$

B. Model Predictive Controller

The dynamics introduced in the previous section are linear in the surrogate state $\mathbf{z}$ and in the control variable, and can be leveraged to define an optimal control problem (OCP), to be solved within an LMPC loop [14]. In this section, we detail our novel MPC strategy. Let $\mathfrak{C}$ be as defined in (12). Furthermore, let $\mathbf{Q} = \mathfrak{C}^\intercal\mathbf{Q}'\mathfrak{C}$, $\mathbf{Q}' \in \mathbb{R}^{3N\times 3N}$, $\mathbf{R} \in \mathbb{R}^{6\times 6}$, and $\boldsymbol{\varphi}_r$ a target pose of the cloth. Lastly, let $\mathcal{U}$ be a convex set containing the admissible control values. Then, we can define the following finite-horizon OCP, to be solved at time-

step κ:

$$\Delta\mathbf{u}_0^*, \dots, \Delta\mathbf{u}_T^* = \\ \underset{\Delta\mathbf{u}_0,\dots,\Delta\mathbf{u}_T}{\arg\min} \sum_{t=0}^{T} \langle(\mathbf{z}_t - \mathbf{z}_r), \mathbf{Q}(\mathbf{z}_t - \mathbf{z}_r)\rangle + \langle\Delta\mathbf{u}_t, \mathbf{R}\Delta\mathbf{u}_t\rangle$$

subject to:

$$\mathbf{z}_0 = (\mathbf{K}_m^\dagger)^{1/2}\mathbf{k}_{m,\boldsymbol{\varphi}_\kappa},$$
$$\mathbf{z}_r = (\mathbf{K}_m^\dagger)^{1/2}\mathbf{k}_{m,\boldsymbol{\varphi}_r},$$
$$\mathbf{z}_{t+1} = (\mathbf{K}_m^\dagger)^{1/2}\mathbf{K}_{mn}\begin{bmatrix} \mathbf{K}_{nm} & \sqrt{n}S_{\mathbf{u}} \end{bmatrix} \cdot \left(\begin{bmatrix} \mathbf{K}_{mn} \\ \sqrt{n}\mathbf{S}_{\mathbf{u}}^T \end{bmatrix}\begin{bmatrix} \mathbf{K}_{nm} & \sqrt{n}\mathbf{S}_{\mathbf{u}} \end{bmatrix} + \gamma n\mathcal{B}(\mathbf{K}_m, I)\right)^\dagger \cdot \mathcal{B}(\mathbf{K}_m^{1/2}, I)\begin{bmatrix} \mathbf{z}_t \\ \mathbf{u}_t \end{bmatrix},$$
$$\Delta\mathbf{u}_t \in \mathcal{U},\ \forall t \in \{0, \dots, T\}. \tag{13}$$

The position at time-step κ to which the grasped points of the cloth need to be moved is obtained as

$$\mathbf{u}_\kappa = \mathbf{u}_{\kappa-1} + \Delta\mathbf{u}_0^*. \tag{14}$$

The control $\mathbf{u}_\kappa$ is applied to the SOM. Then, problem (13) is solved again, with $\boldsymbol{\varphi}_{\kappa+1}$ as initial state, obtained from the SOM.

1) Constraints: The OCP in (13) follows two types of constraints: Those related to the lifting of the state and dynamics, and those constraining the control actions, as described in the following.

a) Dynamics: The first three constraints appearing in (13) correspond to the lifting of the cloth's state at the beginning of the OCP time window, the lifting of the reference state (i.e., the target mesh pose), and the dynamics in the lifted space, obtained through the data-driven approximation of the Koopman operator.

b) Constraining the control: One of the key advantages of MPC is the possibility of naturally embedding constraints in the OCP. This is of upmost importance considering that we are using a data-driven model, i.e., some regions of the control space (and consequently, of the state space) may not have been explored during training, and for trajectory shaping, as we will discuss in the following. Given the definition of the cloth dynamics proposed in Section II, we we propose to shape the control set $\mathcal{U}$ by defining the following constraints:

- The OCP should not involve regions of the control space that were not sampled during the system identification phase, and for which the data-driven system becomes unreliable. Specifically, these constraints are of the form

$$\mathbf{u}_{\kappa-1} + \Delta\mathbf{u}_0 \geq [-\infty, -y_{\text{min}}, h_{\text{min}}]^T \tag{15}$$
$$\mathbf{u}_{\kappa-1} + \Delta\mathbf{u}_0 + \dots + \Delta\mathbf{u}_t \geq [-\infty, -y_{\text{min}}, h_{\text{min}}]^T, \quad \forall t \in \{1, \dots T\}, \tag{16}$$

  to prevent the controlled corner to get too close to (or below) the table the cloth is placed on, along the vertical axis, as well as to fold towards the center of the cloth in the beginning. y_{min} is the initial position of the grasped corner along the y axis, h_{min} is the lowest initial height of the controlled points.
- Smoothness of the generated control trajectories should be enforced. This is of upmost importance considering that the trajectories of the grasped corner of the cloth, generated in simulation, need to be executed by a robotic manipulator, to perform the folding. For $s > 0$, these constraints are of the form

$$-s \leq \Delta\mathbf{u}_0 - (\mathbf{u}_{\kappa-1} - \mathbf{u}_{\kappa-2}) \leq s, \tag{17}$$
$$-s \leq \Delta\mathbf{u}_t - \Delta\mathbf{u}_{t-1} \leq s,\ \forall t \in \{1, \dots, T\}. \tag{18}$$

- The position of the two controlled points in the mesh should vary by the same displacement, to emulate the end-effector of the cloth moving with constant orientation. Let $\Delta\mathbf{u}_t^{[i:j]}$ be the i-to-j entries of $\Delta\mathbf{u}_t$. This constraint can be encoded as

$$\Delta\mathbf{u}_t^{[1:3]} - \Delta\mathbf{u}_t^{[4:6]} = \mathbf{0} \tag{19}$$

- Lastly, we should prevent the grasped corner from being displaced too far from its current position, avoiding unstable behaviors. For $\mathbf{w} \in \mathbb{R}^6$ and $\mathbf{v} \in \mathbb{R}^6$, these constraints are of the form

$$\mathbf{w} \leq \Delta\mathbf{u}_t \leq \mathbf{v},\ \forall t \in \{0, \dots T\}. \tag{20}$$

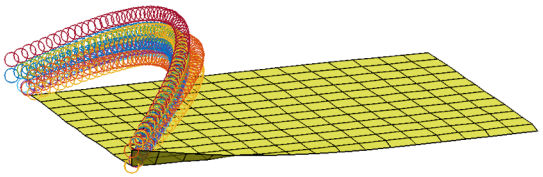


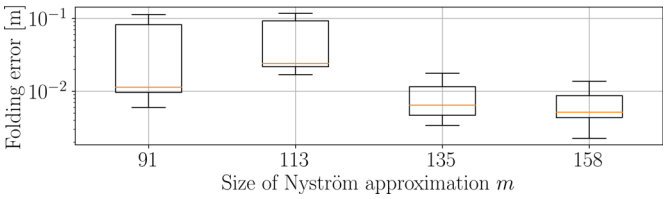
Fig. 3: Some of the training parabolas used for the data generation.

IV. Experiments

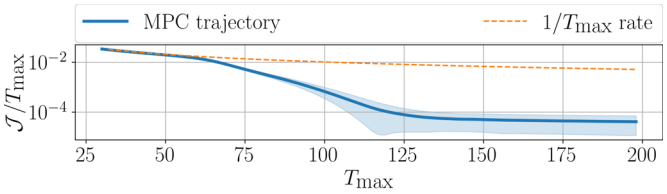
In this section, we evaluate the proposed control pipeline empirically, both in simulation and on a real robotic platform.

A. Simulation Experiments

As a start, we consider an assessment of the proposed LMPC strategy in simulation. The cloth is modeled as a rectangle with size 59 cm $\times$ 42 cm, discretized as a mesh of size 17×13 nodes. The initial flat pose of the cloth is obtained through a motion capture system, as detailed in [21]. As discussed in Section III, our robot manipulates the cloth by keeping a fixed orientation of the end-effector, which is orthogonal to the table the cloth is placed on. We compute the dynamics in (11) by using 100 training trajectories of length 1.5 s, with a sampling time of $\Delta t = 0.01$s. The training folds happen in the form of parabolic trajectories, tilted towards the center of the cloth. While the use of parabolic trajectories introduces a bias in the training data, there is consensus that such trajectory shapes may be effective enough for dynamic folding [22], [7]. Moreover, as we discussed in Section III,

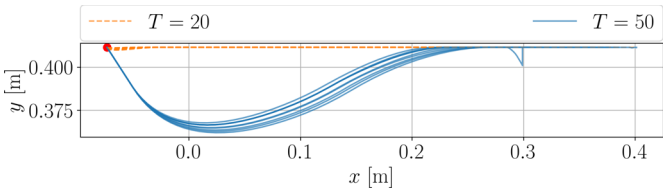


(a) Final folding error.

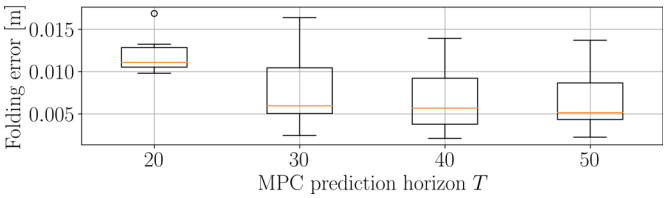


(b) Sublinear cost index $\mathcal{J}$.

Fig. 4: (a) Final folding error evaluated across 10 target folds generated with single-handed motions, for different values of m. Increasing m yields a more accurate COM, which in turn improves the performance of the MPC algorithm. (b) The running cost (21) (normalized in the interval $[0, 1]$), grows at a sublinear rate w.r.t. the horizon $T_{\max}$. The MPC curve is obtained with $m = 158$ Nyström landmarks. Mean $\pm$ std. across 10 target folds.



(a) Control trajectories for horizons T.



(b) Final mesh error $\mathcal{M}$.

Fig. 5: (a) x-y visualization of the trajectories generated with different values of the prediction horizon T, and $m = 158$. An initial motion towards the center of the cloth emerges as a consequence of a larger value for T. This type of motion is pivotal for the fold to be successful. The initial position is marked by a red dot. (b) Final folding error evaluated across 10 target folds generated with single-handed motions, for different values of the prediction horizon T. Increasing T indeed yields a more effective control trajectory.

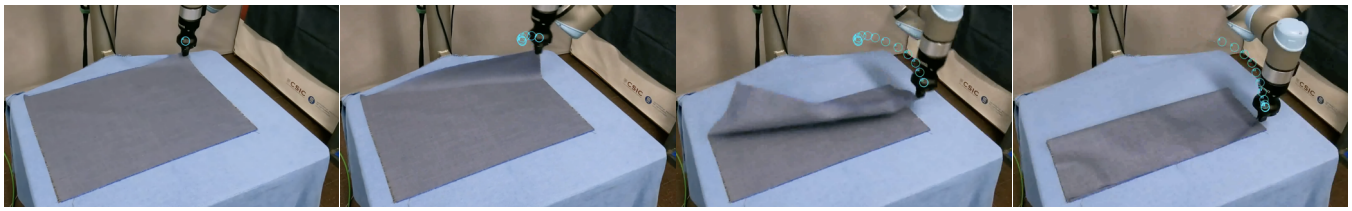


Fig. 6: An illustrative example of a folding trajectory generated by our MPC pipeline, with a woolen piece of cloth. The trajectory performed by the gripper is displayed as numbered cyan circles.

the synthesized MPC trajectories are not constrained to have the shape of parabolas.

In order to simulate the fixed orientation of the end-effector, we simulate the grasping by controlling the corner and the node on the boundary directly adjacent to it. A visualization of some training trajectories is offered in Fig. 3. The simulation pipeline is implemented in `Matlab`, and the OCP is solved online by means of the `YALMIP` library.

Note that, in order to improve the generalization capability of our pipeline, the training set may be augmented by additional data in which, e.g., the initial pose of the cloth is perturbed. On the other hand, if different cloth materials or shapes are employed, the SOM's parameters described in Section II would need to be updated accordingly, our approach being model-based.

1) Impact of the size of the surrogate state: We firstly aim at assessing how the folding error, computed *at the end* of a simulation, varies with m, i.e., the size of the surrogate state $\mathbf{z}$. In this experiment, the MPC prediction horizon is set to 50 time-steps. The folding error is obtained by calculating the Euclidean distance between the meshes, after aligning the target mesh and the one obtained through MPC with a minimal SVD rotation. As shown in Fig. 4, quality of the folds improves for larger values of m.

2) Impact of the prediction horizon: Choosing a large enough size of the surrogate state $\mathbf{z}$, we are now interested on how the size of the MPC prediction window affects the trajectory generation. As illustrated in Fig. 5a, when the horizon is short, the trajectories do not go towards the center of the cloth in the beginning. Conversely, this behavior emerges for larger values of the MPC prediction horizon. We empirically observed that this is a relevant feature for the fold to be successful, as confirmed by Fig. 5b.

3) Running cost: Next, we also consider the performance of the generated trajectories in terms of the following cumulative cost index, computed on the simulation horizon $T_{\max}$, for $Q' = 1\ \mathrm{m}^{-2}$, $R = 500\ \mathrm{m}^{-2}$:

$$\mathcal{J} = \sum_{\kappa=0}^{T_{\max}} (\boldsymbol{\varphi}_\kappa - \boldsymbol{\varphi}_r)^\intercal \mathbf{Q}' (\boldsymbol{\varphi}_\kappa - \boldsymbol{\varphi}_r) + \Delta \mathbf{u}_\kappa^\intercal \mathbf{R} \Delta \mathbf{u}_\kappa. \quad (21)$$

TABLE II: Parameters of the cloths employed in our folding setup.

	Size [cm^2]	Material	Density [kg/m^2]	Example
Cloth I:	59×42	Wool	0.1804	Suit
Cloth II:	59×42	Denim	0.3046	Jeans

This cost can be viewed as the cost our model predictive controller tries to minimize, in a greedy fashion (by means of the receding horizon approach), based on the forecasts of the surrogate model, i.e., $\boldsymbol{\varphi}_t \approx \mathfrak{C}\mathbf{z}_t$, and the constraints on the control action described in Section III. Fig. 4b shows that the cost index $\mathcal{J}$ grows at a sublinear rate compared to the size of the simulation horizon T_{max}[1].

B. Sim-to-Real Evaluation

We can now consider how the trajectories generated with our pipeline transfer to a real robot setup. The goal is again to fold a rectangular piece of cloth by controlling only one of its corners. We consider two different garments, whose parameters can be found in Table II. The chosen materials are representative of common garments that can be encountered in household scenarios, and they differ both in terms of stiffness and density, allowing to test the sim-to-real performance of our algorithm in diverse scenarios. Given the high speed of the motions of interest (with a smaller duration than 1.5 s), the trajectories are executed in open loop, as the processing of high-frequency real-time feedback would give a delay to the controller, i.e.: current state-of-the-art visual feedback techniques for cloth perception would provide data at a low frequency with a high delay, compared to the speed of the motion. Closing the loop would also require additional compensation and simplifications of the model, see, e.g., [23], [24]. The trajectories are executed with a controller in Cartesian space. In this experiment, we train the two data-driven models on 100 trajectories of duration 1.5 s, generated with single-handed motions. The testing folded poses are generated by running 5 bi-manual, quasi-static manipulation trajectories with the SOM. Our experimentation platform is a `UR5` robotic arm that grasps the cloth with a fixed, vertical gripper orientation, as discussed in Section III. Fig. 6 shows an illustrative folding trajectory generated by our MPC pipeline[2].

1) Performance: In order to assess the performance of the generated trajectories, we compute the *folding ratio*, i.e., the ratio between the area of visible surface of the cloth after folding with the real robot, and the area of the unfolded surface. This value, denoted as $\mathcal{R}_{\text{real}}$, is compared against the ratio between the area of the target visible surface from the simulator, and the unfolded surface from the simulator. The latter ratio is denoted as $\mathcal{R}_{\text{target}}$. The trajectories with the real robot are repeated 2 times each. Table III shows the associated key performance indicator (KPI) $\mathcal{E}_{\text{fold}}$, computed as

$$\mathcal{E}_{\text{fold}} = \frac{|\mathcal{R}_{\text{real}} - \mathcal{R}_{\text{target}}|}{\mathcal{R}_{\text{target}}}. \tag{22}$$

TABLE III: Relative folding error $\mathcal{E}_{\text{fold}}$ between the real and target cloth folds according to (22), for 5 target folds. The values are averaged across 2 repetitions of each trajectory.

Material					
Wool	6.02%	3.99%	2.21%	6.44%	7.23%
Denim	3.97%	1.76%	0.769%	4.34%	3.81%

Since this KPI may not be able to capture nuanced features of the achieved fold (e.g., in terms of exact placement of the cloth's corner), we supply a zenithal visualization of such folds in Fig. 7. Indeed, while the denim folds achieve a lower KPI than the woolen ones, we can observe in Fig. 7 that the surface of the wool garment is more flat as opposed to the denim one, and the corner placement is in turn more accurate w.r.t. the reference fold. This is due to the different physical properties of the garments.

Overall, our algorithm achieves a satisfactory sim-to-real performance, being able to fold both garments to a given target pose.

V. Conclusions

In this paper, we have shown how accurate, physics-based modeling of cloth can be used to generate a linear, surrogate model by means of Koopman operator regression. This surrogate model, coupled with suitable constraints, is then embedded in a model predictive control pipeline in closed loop with the physics-aware one, to generate folding trajectories to be executed on a real robot.

Under the assumption that the cloth is initially flat on a table and that we have performed system identification with a physical simulator, the Koopman model has been trained *exclusively on simulated data*, which has been shown here to be realistic enough to achieve a successful zero-shot simulation-to-reality transfer of the generated high-speed trajectories.The non-linear physical model is accurate as it accounts for *aerodynamics* and *inextensibility*. Moreover, the Koopman-based linearization of the cloth's dynamics has a twofold benefit. On one hand, it allows to compute the folding trajectories in a *derivative-free* manner, w.r.t. the non-linear simulator. Moreover, the dimensionality of the cloth's state space is reduced after the Koopman lifting, w.r.t. the original state space. Overall, these points allow to solve efficient linear quadratic optimal control problems.

The model predictive control strategy proposed in this paper opens the way to interesting future research, aimed, for instance, at including feedback from the real cloth status. Such feedback could be employed, e.g., to align the robot with the cloth in an initial phase, and perform *grasping* primitives, so as to start the folding motion according to the simulated environment. In terms of grasping, we further showed that a vertical grasp is enough to achieve successful cloth folds, provided that it is modeled correctly in the

[1]Note that, in our implementation, the MPC is started at $T_{\text{max}} = 30$, in order for the SOM to reach a stable initial state.

[2]Gripper tracking realized with the `Tracker` software.

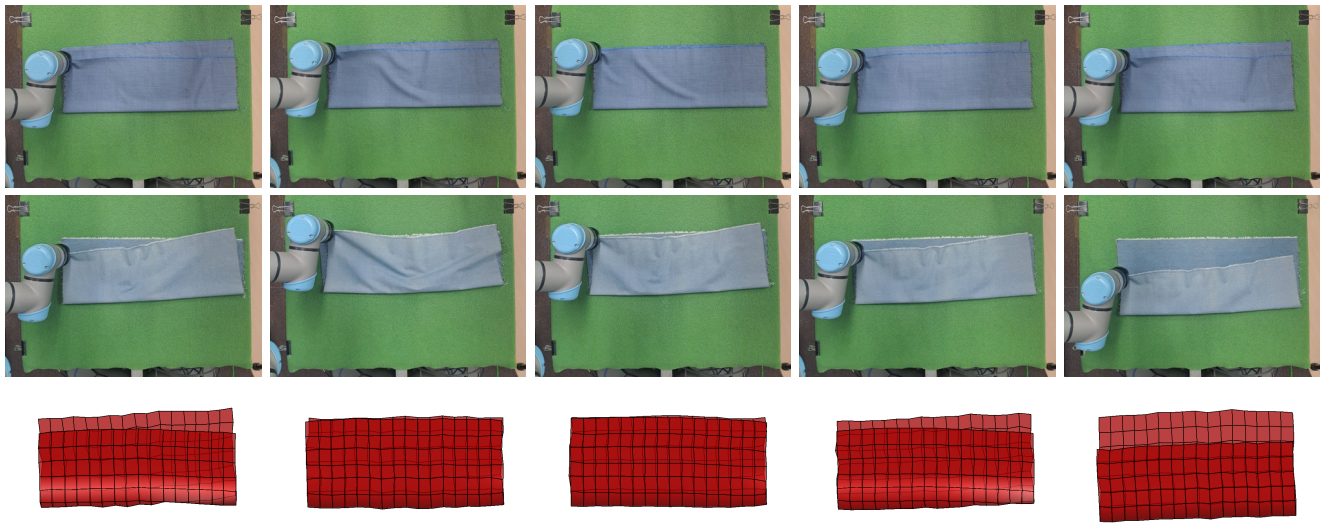


Fig. 7: A qualitative visualization of the folds obtained with our MPC pipeline, in the sim-to-real evaluation. The top row contains the results for wool, the second row the results for denim, and the bottom row the target folds.

simulator. Nonetheless, different types of grasps could be studied and developed, to account for different cloth shapes and parameters.

REFERENCES

[1] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[2] A. Longhini, Y. Wang, I. Garcia-Camacho, D. Blanco-Mulero, M. Moletta, M. Welle, G. Alenyà, H. Yin, Z. Erickson, D. Held, *et al.*, "Unfolding the literature: A review of robotic cloth manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, 2024.

[3] F. Coltraro, J. Amorós, M. Alberich-Carramiñana, and C. Torras, "An inextensible model for the robotic manipulation of textiles," *Applied Mathematical Modelling*, vol. 101, pp. 832–858, 2022.

[4] F. Coltraro, J. Amorós, M. Alberich-Carramiñana, and C. Torras, "A novel collision model for inextensible textiles and its experimental validation," *Applied Mathematical Modelling*, vol. 128, pp. 287–308, 2024.

[5] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robot Learning*, pp. 24–33, PMLR, 2022.

[6] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, IEEE, 2022.

[7] J. Hietala, D. Blanco-Mulero, G. Alcan, and V. Kyrki, "Learning visual feedback control for dynamic cloth folding," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1455–1462, IEEE, 2022.

[8] A. Longhini, M. C. Welle, Z. Erickson, and D. Kragic, "Adafold: Adapting folding trajectories of cloths via feedback-loop manipulation," *IEEE Robotics and Automation Letters*, 2024.

[9] M. T. Mason and K. M. Lynch, "Dynamic manipulation," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 1, pp. 152–159, IEEE, 1993.

[10] D. Blanco-Mulero, G. Alcan, F. J. Abu-Dakka, and V. Kyrki, "QDP: Learning to sequentially optimise quasi-static and dynamic manipulation primitives for robotic cloth manipulation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 984–991, IEEE, 2023.

[11] D. Blanco-Mulero, O. Barbany, G. Alcan, A. Colomé, C. Torras, and V. Kyrki, "Benchmarking the sim-to-real gap in cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2981–2988, 2024.

[12] E. Caldarelli, A. Chatalic, A. Colomé, C. Molinari, C. Ocampo-Martinez, C. Torras, and L. Rosasco, "Linear quadratic control of nonlinear systems with Koopman operator learning and the Nyström method," *Automatica*, vol. 177, p. 112302, 2025.

[13] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[14] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[15] P. Bevanda, S. Sosnowski, and S. Hirche, "Koopman operator dynamical models: Learning, analysis and control," *Annual Reviews in Control*, vol. 52, pp. 197–212, 2021.

[16] F. Coltraro, J. Amorós, C. Torras, and M. Alberich-Carramiñana, "A practical aerodynamic model for dynamic textile manipulation in robotics," *Mechanism and Machine Theory*, vol. 209, p. 105993, 2025.

[17] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, and E. Grinspun, "A quadratic bending model for inextensible surfaces," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pp. 227–230, Eurographics Association, 2006.

[18] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals, Sixth Edition*. Butterworth-Heinemann, May 2005.

[19] E. J. Nyström, "Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben," 1930.

[20] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," *Advances in neural information processing systems*, vol. 13, 2000.

[21] F. Coltraro, J. Borràs, M. Alberich-Carramiñana, and C. Torras, "Tracking cloth deformation: A novel dataset for closing the sim-to-real gap for robotic cloth manipulation learning," *The International Journal of Robotics Research*, p. 02783649251317617, 2025.

[22] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, "Folding deformable objects using predictive simulation and trajectory optimization," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6000–6006, IEEE, 2015.

[23] A. Luque, D. Parent, A. Colomé, C. Ocampo-Martinez, and C. Torras, "Model predictive control for dynamic cloth manipulation: Parameter learning and experimental validation," *IEEE Transactions on Control Systems Technology*, 2024.

[24] E. Caldarelli, A. Colomé, C. Ocampo-Martinez, and C. Torras, "Quadratic dynamic matrix control for fast cloth manipulation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8178–8185, IEEE, 2023.