

Adapting the Servo-constraints Model Inversion Method for Robotic Cable Manipulation

Louis Dehaybe, Olivier Brls

Aerospace and Mechanical Engineering Department
University of Liège
Allée de la Découverte 9, 4000 Liège, Belgium
(louis.dehaybe, o.bruls)@uliege.be

Introduction

Robotic Deformable Object Manipulation (DOM) is a currently active research topic, highly relevant in industrial, surgical, and domestic settings. We consider one or more robots firmly grasping a deformable object (DO) at arbitrary locations. Given the current DO configuration $\mathbf{q}(0) \in \mathbb{R}^{n_q}$ and the corresponding output $\mathbf{z} = \mathbf{h}(\mathbf{q}) \in \mathbb{R}^{n_z}$, we seek the robot control inputs $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ (i.e., the gripper motions) required to reach a desired output state $\mathbf{z}_d(t_f) = \mathbf{h}(\mathbf{q}_d(t_f))$, where time $t \in [0, t_f]$. Previously [1], we proposed a pure feedback-based and model-free solution using online Jacobian estimation. This relies on several assumptions, such as quasi-static manipulations, continuous feedback from cameras (often hindered by occlusions) and, ideally, persistent excitations in all input directions. To alleviate these limitations, this work explores incorporating model-based knowledge into manipulation algorithms. In [5], the DOM task is formulated as a *trajectory optimization* problem: the trajectory to follow $\mathbf{q}_d(t)$ is not known a priori but determined and optimized together with the control inputs $\mathbf{u}(t)$. In contrast, we consider a two-stage approach: first, a planner generates a desired trajectory $\mathbf{z}_d(t)$; second, a *trajectory tracker* computes the control inputs $\mathbf{u}(t)$ such that the object deforms as prescribed. The rationale is that sampling-based planners used in robotics (e.g., RRTs, see [4]) are usually more efficient at avoiding local minima than optimization-based methods. This work addresses the second stage: we consider $\mathbf{z}_d(t)$ as given in order to focus on trajectory tracking, which we propose to tackle via inverse dynamics and the servo-constraints approach. We specifically consider cables manipulation, as illustrated by two examples in Fig. 1.

Methodology

For the sake of generality, we model cables as flexible multibody systems, so that the equations of motion take the form of differential-algebraic equations:

$$\dot{\mathbf{q}} = \mathbf{c}(\mathbf{q}, \mathbf{v}) \quad (1)$$

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} + \mathbf{f}(\mathbf{q}, \mathbf{v}, t) + \mathbf{G}^T(\mathbf{q})\boldsymbol{\lambda} = \mathbf{B}\mathbf{u} \quad (2)$$

$$\mathbf{g}(\mathbf{q}) = \mathbf{0}_{n_g \times 1} \quad (3)$$

where $\mathbf{v} \in \mathbb{R}^{n_q}$ is the velocity vector, $\mathbf{f} \in \mathbb{R}^{n_q}$ gathers internal and inertia forces, $\mathbf{g} \in \mathbb{R}^{n_g}$ is the vector of kinematic constraints with gradient matrix $\mathbf{G} \in \mathbb{R}^{n_g \times n_q}$, $\mathbf{M} \in \mathbb{R}^{n_q \times n_q}$ is the mass matrix, $\boldsymbol{\lambda} \in \mathbb{R}^{n_g}$ are the Lagrange multipliers, and $\mathbf{B} \in \mathbb{R}^{n_q \times n_u}$ is the input distribution matrix. Equation (1) represents the kinematic compatibility condition.

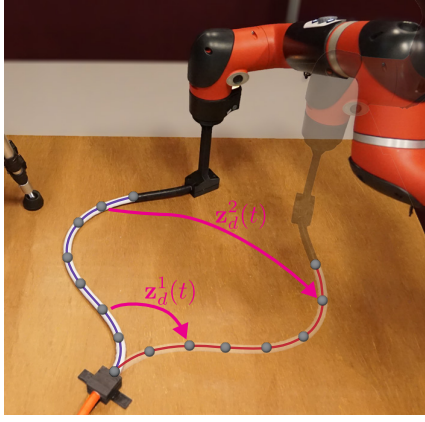
As detailed in [2], various model inversion methods exist. For instance, the well-known computed torque method is an obvious choice to solve inverse dynamics problems in robotics. However, this is restricted to fully-actuated systems, while DOM is fundamentally heavily underactuated. Therefore, we investigate two alternative directions. The first approach is to use *optimal control* (OC) methods: minimize a cost function representing the tracking error over the full trajectory, for instance:

$$\min_{\mathbf{u}} \int_0^{t_f} \frac{1}{2} (\mathbf{h}(\mathbf{q}) - \mathbf{z}_d)^T (\mathbf{h}(\mathbf{q}) - \mathbf{z}_d) dt \quad (4)$$

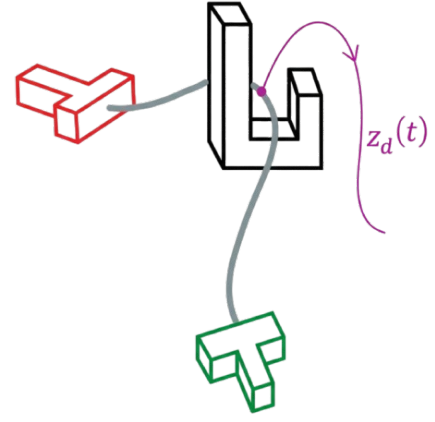
with Eqs. (1)-(3) acting as constraints to the optimization problem. The second approach, also explored in [2, 3], is to supplement Eqs. (1)-(3) by the so-called *servo-constraints* (SC)

$$\mathbf{h}(\mathbf{q}(t)) - \mathbf{z}_d(t) = \mathbf{z}(t) - \mathbf{z}_d(t) = \mathbf{0} \quad (5)$$

Equations (1)-(3),(5) are solved to determine the feedforward control input $\mathbf{u}(t)$. Note that, as shown in [3], the approach can be extended to the case where $\mathbf{q}$ does not belong to a vector space but includes elements of $SE(3)$, for instance.



(a)



(b)

Figure 1: (a) Robotic cable manipulation in 2D. The position of the grey dots on the cable constitute the output $\mathbf{z}(\mathbf{t})$. The desired trajectory of two of them is illustrated by $\mathbf{z}_d^1(t)$ and $\mathbf{z}_d^2(t)$. (b) Conceptual representation of a 3D task: two grippers (in green and red) must remove a cable from a hook (in black) by controlling a point to follow the trajectory $\mathbf{z}_d(t)$.

A first research question addressed in this work concerns the choice of appropriate output mappings $\mathbf{h}(\mathbf{q})$. For instance, one could use a set of discrete points (as in Fig. 1(a)) or frames distributed on the object (both leading to high n_z), or discrete strain values (leading to lower n_z).

Secondly, we apply the trajectory tracking approaches to DOM. The SC method is simple and computationally efficient. However, it requires $n_z = n_u$, which is hard to satisfy in this context. For instance, representing a cable using the position of multiple points [1] leads to $n_z \gg n_u$. Furthermore, $\mathbf{z}_d(t)$ should not conflict with the kinematic constraints (3) nor impose conflicting motions (e.g., $\mathbf{z}_d^1(t)$ and $\mathbf{z}_d^2(t)$ must be simultaneously feasible in Fig. 1(a)) [2]. However, standard DOM planners frequently employ simplified models that address only the geometric problem. They may also ignore the control authority the robot has on the object's shape (which is grasp-dependent). This means that the error can not always be effectively reduced to zero. While OC handles these discrepancies by finding a "best-fit" solution that minimizes the cost function, standard SC methods may fail to return any solution. To resolve this, we propose to modify the SC with a projection operator $\mathbf{P} \in \mathbb{R}^{n_u \times n_z}$ such that they become $\mathbf{P}(\mathbf{z}(t) - \mathbf{z}_d(t)) = \mathbf{0}$. By projecting the tracking error into a lower space, we ignore the orthogonal components of the error that the system cannot physically correct or which might require unbounded control inputs. This modification allows for better workflows combining the inversion method and higher-level planning tools. It also provides more flexibility in the representation of the object, allowing $n_u > n_z$.

In summary, our contributions include: (i) applying the SC method to robotic DOM, (ii) proposing a modification of the servo constraints by introducing a projection operator, (iii) comparing inverse dynamics results obtained with the modified SC against OC solutions in practical cable manipulation tasks.

References

- [1] Dehaybe, L.; Bröls, O.: Enhancing Robotic Shape Servoing Methods Through an SE(3)-Based Representation of Shapes and Deformations. Available at SSRN 5387664, 2024.
- [2] Drücker, S.; Seifried, R.: Trajectory-tracking control from a multibody system dynamics perspective. *Multibody System Dynamics*, Vol. 58, No. 3, pp. 341–363, 2023.
- [3] Lismonde, A.; Sonnevile, V.; Bröls, O.: A geometric optimization method for the trajectory planning of flexible manipulators. *Multibody System Dynamics*, Vol. 47, No. 4, pp. 347–362, 2019.
- [4] Yu, M.; Lv, K.; Wang, C.; Jiang, Y.; Tomizuka, M.; Li, X.: Generalizable whole-body global manipulation of deformable linear objects by dual-arm robot in 3-d constrained environments. *The International Journal of Robotics Research*, Vol. 44, No. 4, pp. 607–639, 2025.
- [5] Zimmermann, S.; Poranne, R.; Coros, S.: Dynamic manipulation of deformable objects with implicit integration. *IEEE Robotics and Automation Letters*, Vol. 6, No. 2, pp. 4209–4216, 2021.