Data-driven learning of the manifolds of a robotic manipulator

Jérôme Adriaens, Guillaume Drion, Pierre Sacré
Dept. of Electrical Engineering and Computer Science, University of Liège, Belgium
jadriaens@uliege.be, gdrion@uliege.be, p.sacre@uliege.be

1 Introduction

Recently, many hand-like robotic manipulators with high degrees of freedom have been developed. Specifically, anthropomorphic robotic hand models such as the Shadow Hand by Shadow Robot are highly redundant and offer significant potential for effective object manipulation. However, controlling such systems is difficult due to the high dimensionality of the joint space [1]. Paradoxically, those kinds of manipulations are deemed easy for humans who do not control each joint of their hand individually but rather use a low-dimensional representation of their hand to perform the manipulation often referred to as synergies [2].

In this work, we propose to learn the manifolds in the joint space resulting from the synergies via a robotic manipulator using an autoencoder. We will see how the learned manifolds can be analyzed and used to simplify the control of the manipulator.

2 Generalized coordinates manifold mapping

First of all, we state the problem from a rigid body dynamics point of view. For a particular robotic manipulator, we have the following dynamics equations:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau, \tag{1}$$

where q are the joint angles, $\dot{q}$ the joint velocities, $\ddot{q}$ the joint accelerations, τ the joint torques, $M(q)$ the mass matrix, $C(q,\dot{q})$ the Coriolis matrix, and $G(q)$ the gravity vector. Usually, we define a mapping between the low-dimensional space of the generalized coordinates $\mathscr{S}$ to the full dimensional space of the joint angles $\mathscr{Q}$:

$$h:\mathscr{S}\to\mathscr{Q}. \tag{2}$$

We can see $\mathscr{S}$ as the space of the manifold coordinates that lies in the space $\mathscr{Q}$. Our goal is to learn that manifold $\mathscr{S}$ through h using collections of data obtained from manipulators in execution.

3 Learning the manifold

To learn the manifold $\mathscr{S}$, we propose to use an autoencoder, a now common non-linear reduction technique. The autoencoder comprises two main parts: an encoder and a decoder. The encoder maps the high-dimensional space $\mathscr{Q}$ to the low-dimensional space $\mathscr{S}$, while the decoder maps the other way around. Denoting the encoder as E_θ and the decoder as D_ϕ, with θ and ϕ the parameters of the encoder and decoder respectively, we have

$$E_\theta:\mathscr{Q}\to\mathscr{S} \tag{3}$$

$$D_\phi:\mathscr{S}\to\mathscr{Q} \tag{4}$$

In our case, D_ϕ will act as the mapping h defined in (2) and, ideally, we want $D_\phi=E_\theta^{-1}$. During the optimization process of θ and ϕ, we want to minimize the difference between the input and the output of the autoencoder while remaining consistent with the topology of the robotic manipulator. Drawing inspiration from physics-informed machine learning [3], [4], the desired consistency is constrained using the mass matrix $M(q)$ and the Jacobian matrix of the robotic manipulator linking the joint space and the cartesian space.

Once the optimization process is completed, we can analyze the learned manifold $\mathscr{S}$ and the dynamics of the robotic manipulator on that manifold.

4 Application

Eventually, we want to apply this method to the anthropomorphic hand from Shadow Robot. The data from this particular manipulator are collected by the use of a teleoperation system with a human operator.

We will illustrate our proposed method and look at the emerging manifolds on simple tasks as a proof of concept and then, on the robotic hand.

Finally, our approach is meant to be used to control a robotic hand by using only a few degrees of freedom.

References

[1] OpenAI, I. Akkaya, M. Andrychowicz, *et al.*, *Solving Rubik's Cube with a Robot Hand*, 2019.

[2] M. L. Latash, "Synergic control of movement: From single muscles to the whole body," in *Progress in Motor Control*, Elsevier, 2024, pp. 25–47.

[3] M. Cranmer, S. Greydanus, S. Hoyer, *et al.*, *Lagrangian Neural Networks*, 2020.

[4] S. Xiao, J. Zhang, and Y. Tang, *Generalized Lagrangian Neural Networks*, 2024.

Acknowledgment This work was supported by the Belgian Government through the FPS Policy and Support (BOSA) grant NEMODEI.