Consensus-based Normalizing-Flow Control: A Case Study in Learning Dual-Arm Coordination

Hang Yin[1], Christos K. Verginis[2] and Danica Kragic[1]

Abstract— We develop two consensus-based learning algorithms for multi-robot systems applied on complex tasks involving collision constraints and force interactions, such as the cooperative peg-in-hole placement. The proposed algorithms integrate multi-robot distributed consensus and normalizing-flow-based reinforcement learning. The algorithms guarantee the stability and the consensus of the multi-robot system's generalized variables in a transformed space. This transformed space is obtained via a diffeomorphic transformation parameterized by normalizing-flow models that the algorithms use to train the underlying task, learning hence skillful, dexterous trajectories required for the task accomplishment. We validate the proposed algorithms by parameterizing reinforcement learning policies, demonstrating efficient cooperative learning, and strong generalization of dual-arm assembly skills in a dynamics-engine simulator.

I. INTRODUCTION

From heterogenous robot teams to anthropomorphic humanoids, modern robot systems are envisioned to enable complex tasks, such as dual-arm manipulation [1], considering multiple robots and end-effectors. Serving as an inspiration, humans master bimanual skills ranging from transporting large objects to fine watch-making tasks [2]. To achieve human-level proficiency, dual-arm manipulation entails development of skillful and dexterous trajectories to bring objects to desired poses, avoid intra- and inter-arm collisions and accommodate physical contacts for stable force interaction. Related work relies on assumptions of established grasps, focuses on resolving task priorities [3], [4] or regulating internal stress [5], [6]. The advancement on learning and synthesizing dexterous motion trajectories for dual- and multi-arm robot systems is still rather limited.

Prior works on trajectory representation and learning have extensively used dynamical-system-based approaches [7], [8]. Ongoing research on robot learning concentrates almost exclusively on the convergence of single-robot systems to fixed point [9], [10]. These well-studied approaches are not directly applicable in multi-robot tasks, since the latter require accommodation of the inter-robot dynamics and the stability of the relative robot poses. These challenges call for learning of dynamical systems on multiple robot agents.

From a control perspective, numerous works use distributed consensus protocols for multi-robot coordination [11], [12], [13]. Each robotic agent uses local information from its neighboring robots to update its internal state, such as its position, and reach an agreement with the team. However, standard consensus protocols cannot produce complex and nonlinear transient behaviors for the robotic agents; such behaviors are necessary, e.g. for the accomplishment of the dual-arm assembly task illustrated in Fig. 1. Apart from achieving specific relative poses, such tasks require skillful trajectories that avoid collisions and accommodate force interactions. Although dual-arm robots are often deployed for such tasks, a multi-robot scenario may deploy independent robotic agents, such as mobile manipulators. Thus, scalability may require distributed algorithms, where the robots use local information to compute their actions.

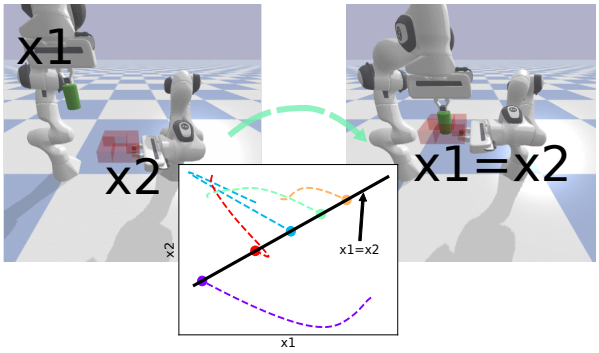


Fig. 1: Consensus-based normalizing-flow for nonlinear control of complex state trajectories (colored dotted lines) with convergence towards a subspace (black bold line) instead of a fixed point. Using dual-arm assembly as an illustrative case.

We study the multi-robot coordination of complex tasks relying on relative pose control. Our contribution is the development of multi-robot algorithms that integrate reinforcement learning and consensus-based stabilization. We design two distributed control algorithms that guarantee the consensus of a multi-robot system with 2nd-order continuous-time dynamics based on a diffeomorphic mapping. The first algorithm achieves consensus among an arbitrary number of robots with standard normalizing-flow networks as the diffeomorphic mapping. The second algorithm encodes translation-invariant control for representing dual-arm coordination strategies that are invariant in the workspace. This is enabled by a modified normalizing-flow model which constitutes another contribution. We study the efficiency of the

[1]H. Yin and D. Kragic are with CAS/RPL, KTH Royal Institute of Technology, Stockholm, Sweden. email:{hyin,dani}@kth.se

[2]C. K. Verginis is with the Division of Signals and Systems, Department of Electrical Engineering, Uppsala University, Uppsala, Sweden. email: christos.verginis@angstrom.uu.se

proposed algorithms in a simulated dual-arm assembly task, where reinforcement learning with a baseline deep neural network policy fails. Our results show that the proposed controllers can learn the desired skills efficiently and generalize them to untrained workspace positions.

II. RELATED WORK

Multi-robot consensus: Research on multi-robot consensus focuses on the development of distributed control algorithms such that the robotic agents of a multi-robot system reach an agreement over some internal variable, such as their positions [11], [13]. Most works model the multi-robot communication via graphs, considering both directed [14], [15] and undirected cases [11]. Other variations regarding multi-robot consensus include the incorporation of a leader agent that excites the system [16], [17], design of observers to account for insufficient measurements [18], or event-triggered communication and control [19], [20]. Finally, a large variety of consensus-based works deals with uncertainties in the agents' dynamics [15], [21].

Learning with dynamical systems: Learning robot task motion commonly uses dynamical systems to represent and generate complex state trajectories. Dynamic Movement Primitives encode stable point-to-point or rhythmic motions with time-dependent dynamical systems [7]. Other works use Gaussian Mixture Models as time-invariant dynamical systems and employ imitation learning of parameters in a constrained space [8]. Time-invariant stable dynamical systems are also explored in reinforcement learning with variable impedance controllers [22] and deep energy models [23]. The authors of [24] propose the use of diffeomorphic transformations for accurate modeling of more complex motions. The concept is the coordinate change of stable linear systems via nonlinear diffeomorphic transformations, whose parameter space is less restricted by the stability conditions. Recent research models such transformations using normalizing-flow neural networks [25], [26] with an unconstrained parameter space [9]. A similar idea is extended to account for second-order dynamics and force control, enabling efficient reinforcement learning in contact-rich manipulation [10]. Normalizing-flow dynamical systems have also been used in computer graphics for learning and generating human locomotive behaviors [27], [28].

Our work uses normalizing-flow neural networks to transform linear controllers with reserved theoretical guarantees. We consider consensus in the context of multi-robot dynamics, which are less explored in terms of learning time-invariant dynamical systems. As opposed to the aforementioned works that focus on learning independent point-to-point behaviors, this paper concerns stable trajectories on forming relative poses; the latter can be more efficient for dual-arm tasks entailing coordination of inter-robot dynamics.

Dual-arm and cooperative manipulation: Research on dual- and multi-arm manipulation focuses largely on optimal load distribution and path tracking for grasped objects [1], [4], [5]. Most works focus on developing decentralized control algorithms to guarantee that a rigidly grasped object tracks a pre-defined trajectory [29], [30], [31]. The works [32], [33] consider similar frameworks for non-rigid grasps, such as rolling contacts. On a different direction, the works [34], [6], [35], [36] focus on the force decomposition problem; that is, how to decompose a desired force, which is to be applied to a grasped object, to individual robot forces such that the total internal stress is zero.

Another important topic is coordination of the relative motion between robots, such as in reaching to grasp or manipulating non-rigid objects. The work [37] follows a master-slave scheme to synchronize kinematic trajectories towards a moving virtual target. The trajectories are generated from linear parameter-varying systems with parameter constraints similar to [8] and an external allocation variable for synchronization. [38] exploits an extended cooperative task-space representation to develop relative motion without intervening the global motion of an articulated object. Research in [39] tackles deformable shape control by precomputing valid dual-arm poses associated to an elastic rod. Finally, [40] addresses learning to align two parts under a master-slave framework. The proposed methodology relies on choosing a proper center of compliance and estimating linear impedance parameters from tele-operated demonstrations.

Our work proposes distributed controllers at the force level while learning complex multi-robot behaviors through parameterized normalizing-flow neural networks in an unconstrained parameter space. We consider a dual-arm assembly task as a case study; in contrast to [40], however, the proposed controllers guarantee multi-robot consensus and the unconstrained parameterization enables the robots to autonomously acquire motion trajectories, e.g. through reinforcement learning, without resorting to demonstrations as in [40] and [37].

III. PRELIMINARIES AND PROBLEM FORMULATION

A. Graph Theory and Multi-Robot Consensus

An undirected graph is a pair $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, where $\mathscr{N}$ is a finite set of nodes, representing a team of robotic agents, and $\mathscr{E} \subset \mathscr{N} \times \mathscr{N}$, with $(i,i) \notin \mathscr{E}$, is the set of edges that model the communication capabilities among the agents. The adjacency matrix associated with the graph $\mathscr{G}$ is denoted by $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$, with $a_{ij} \in \{0,1\}$, $i,j \in \{1,\dots,n\}$. If $a_{ij} = 1$, then agent i obtains information regarding the state x_j of agent j (i.e., $(i,j) \in \mathscr{E}$), whereas if $a_{ij} = 0$ then there is no state-information flow from agent j to agent i (i.e., $(i,j) \notin \mathscr{E}$). Furthermore, the set of neighbors of agent i is denoted by $\mathscr{N}_i = \{j \in \mathscr{N} : (i,j) \in \mathscr{E}\}$, and the degree matrix is defined as $\mathbf{D} =$

$\text{diag}\{|\mathcal{N}_1|,\ldots,|\mathcal{N}_n|\}$. Since the graph is undirected, the adjacency is a mutual relation, i.e., $a_{ij}=a_{ji}$, rendering $\mathbf{A}$ symmetric. The Laplacian matrix of the graph is defined as $\mathbf{L}=\mathbf{D}-\mathbf{A}$, which is symmetric and positive semidefinite [13]. The graph is connected if there exists a path between any two agents. For a connected graph, the kernel of $\mathbf{L}$ is $\text{span}\{\mathbf{1}\}$, where $\mathbf{1}$ is the vector of ones of appropriate dimension, implying that $\mathbf{L1}=0$ [13].

The distributed multi-robot consensus problem consists of the agreement of the state trajectories $x_i(t)$, $i\in\mathcal{N}=\{1,\ldots,n\}$, of a team of n robotic agents using local information. More specifically, each agent updates its state based on the first-order protocol $\dot{x}_i=-\sum_{j\in\mathcal{N}_i}a_{ij}(x_i-x_j)$, which uses local information from the neighboring set $\mathcal{N}_i$. The aforementioned update is equivalent to $\dot{x}=-\mathbf{L}x$ in stacked vector form, with $x=[x_1,\ldots,x_n]^\top$. If the multirobot graph is connected, then x converges to the kernel of $\mathbf{L}$, which is $\text{span}\{\mathbf{1}\}$, implying that $\lim_{t\to\infty}(x_i(t)-x_j(t))=0$, for $i,j\in\mathcal{N}$, $i\neq j$ [13]. Numerous works extend the consensus protocol for higher-order systems [41].

B. Normalizing-Flow Neural Networks

Normalizing-flow models are invertable neural networks that are frequently used in learning probabilistic generative models [26]. The models are constructed by a sequence of bijective and differentiable layers with parameterized neural networks. Such bijections allow the transformation of the likelihood of a simple probability distribution, such as Gaussian, into more complex ones through the respective Jacobian terms (Fig. 2 left). As a result, density estimation can be performed by maximizing a tractable data likelihood. Differentiable bijections are also used as diffeomorphic mappings to transform state spaces that can characterize complex trajectories [9], [10]. An example is depicted in Fig. 2; the right part shows that mapping the original state coordinate can warp a linear 2D gradient field to model complex flows. We exploit such a property in the next section to establish consensus guarantees for multi-robot systems.

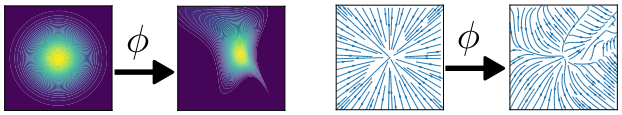


Fig. 2: Normalizing-flow networks transform simple density functions or vector fields to complex models, with tractability or dynamic properties retained. Adapted from [10].

A specific normalizing-flow model called RealNVP [42] parameterizes invertible layers with the following equations (also see Fig. 3):

$$\begin{aligned}\mathbf{z}^{k+1}&=\boldsymbol{\phi}^k(\mathbf{z}^k)\\ \mathbf{z}^{k+1}_{1:d^k}&=\mathbf{z}^k_{1:d^k}\\ \mathbf{z}^{k+1}_{d^k+1:d}&=\mathbf{z}^k_{d^k+1:d}\odot\exp(\mathbf{s}^k(\mathbf{z}^k_{1:d^k}))+\mathbf{t}^k(\mathbf{z}^k_{1:d^k})\end{aligned}\tag{1}$$

where $\mathbf{z}^k$ is the input to the layer, D is the dimension of the input and $d^k<D$ is the index where the input dimensions are partitioned at layer k. The partitions are coupled through elementwise multiplication $\odot$ and nonlinear transformations $\mathbf{s}^k(*)$ and $\mathbf{t}^k(*)$ which are ordinary neural networks. Stacking these layers yields the entire model $\boldsymbol{\phi}_{\boldsymbol{\theta}}=\boldsymbol{\phi}^k\circ\boldsymbol{\phi}^{k-1}\circ\ldots\circ\boldsymbol{\phi}^1$ with the invertibility cascaded. The subscript $\boldsymbol{\theta}$ denotes all neural network parameters and is omitted when it is obvious. Note that normalizing-flows like RealNVP is not equivariant in general. However, the way of its nonlinear coupling allows a simple extension for antipodal-equivariance $\boldsymbol{\phi}(\mathbf{z})=-\boldsymbol{\phi}(-\mathbf{z})$, see Section IV-C. We will use this extension in one of our theoretical results.

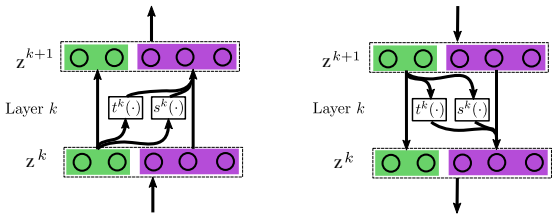


Fig. 3: RealNVP Normalizing-flow The invertible layers wire partitioned dimensions with affine transformations [42]. Adapted from [10].

C. Problem Formulation

We consider $n\geq 2$ robotic agents, characterized by their generalized coordinate $\mathbf{x}_i\in\mathbb{R}^d$, $i\in\mathcal{N}=\{1,\ldots,n\}$. The dynamics of the ith agent can be represented as

$$\mathbf{M}_i(\mathbf{x}_i)\ddot{\mathbf{x}}_i+\mathbf{C}_i(\mathbf{x}_i,\dot{\mathbf{x}}_i)\dot{\mathbf{x}}_i+\mathbf{g}_i(\mathbf{x}_i)=\boldsymbol{\tau}_i+\boldsymbol{\tau}_{i,\text{ext}}\tag{2}$$

where $\mathbf{M}_i:\mathbb{R}^d\to\mathbb{R}^{d\times d}$ is the positive definite inertia matrix, $\mathbf{C}_i:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}^{d\times d}$ is the Coriolis matrix, and $\mathbf{g}_i:\mathbb{R}^d\to\mathbb{R}^d$ is the gravitational force. The term $\boldsymbol{\tau}_i\in\mathbb{R}^d$ represents the actuation of each robotic agent, whereas $\boldsymbol{\tau}_i^{\text{ext}}\in\mathbb{R}^d$ is an external generalized force $\boldsymbol{\tau}_i^{\text{ext}}\in\mathbb{R}^d$. The inertia and Coriolis terms satisfy the skew-symmetric property $\mathbf{y}^\top[\dot{\mathbf{M}}(\mathbf{x}_i)-2\mathbf{C}_i(\mathbf{x},\dot{\mathbf{x}}_i)]\mathbf{y}=0$, for all vectors $\mathbf{y}\in\mathbb{R}^d$, and $i\in\mathcal{N}$.

We further define the stack terms $\mathbf{x}=[\mathbf{x}_1^\top,\ldots,\mathbf{x}_n^\top]^\top\in\mathbb{R}^{nd}$, $\mathbf{M}(\mathbf{x})=\text{diag}\{\mathbf{M}_1(\mathbf{x_1}),\ldots,\mathbf{M}_n(\mathbf{x}_n)\}$, $\mathbf{C}(\mathbf{x})=\text{diag}\{\mathbf{C}_1(\mathbf{x_1},\dot{\mathbf{x}}_1),\ldots,\mathbf{C}_n(\mathbf{x_n},\dot{\mathbf{x}}_n)\}$, $\mathbf{g}(\mathbf{x})=[\mathbf{g}_1(\mathbf{x}_1)^\top,\ldots,\mathbf{g}_n(\mathbf{x}_n)^\top]^\top$, $\boldsymbol{\tau}=[\boldsymbol{\tau}_1^\top,\ldots,\boldsymbol{\tau}_n^\top]^\top$, and $\boldsymbol{\tau}_{\text{ext}}=[\boldsymbol{\tau}_{1,\text{ext}}^\top,\ldots,\boldsymbol{\tau}_{n,\text{ext}}^\top]^\top$, which gives rise to the multi-agent dynamics

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}}+\mathbf{C}(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}}+\mathbf{g}(\mathbf{x})=\boldsymbol{\tau}+\boldsymbol{\tau}_{\text{ext}}.\tag{3}$$

We model the communication among the robotic agents via an undirected and static graph $\mathcal{G}=(\mathcal{N},\mathcal{E})$, as elaborated in Section III-A.

The goal of this paper is learning distributed controllers of the form $\boldsymbol{\tau}=\boldsymbol{\pi}(\mathbf{x},\dot{\mathbf{x}},\boldsymbol{\theta})$ that guarantee multi-robot consensus in a transformed, diffeomorphic space. Such a space is dictated by a normalizing-flow neural network $\boldsymbol{\phi}$ and parameterized by $\boldsymbol{\theta}$, as explained in

Section III-B. As opposed to traditional multi-robot consensus protocols, the proposed approach allows the robotic agents to learn how to execute skillful trajectories that accomplish a complex multi-robot task. For instance, consider the dual-arm assembly example of Fig. 1, where two robotic agents coordinate to achieve a peg-in-hole task. The developed controllers achieve consensus on the two parts, i.e., $\mathbf{x}_1 = \mathbf{x}_2$, while the agents learn, via a normalizing-flow neural network $\boldsymbol{\phi}(\cdot)$ and a reinforcement-learning procedure, how to execute trajectories that avoid collisions and accommodate force interactions during insertion.

IV. MAIN RESULTS

This section presents the main theoretical results of the paper. Firstly, we present a distributed nonlinear controller for the general case of n robotic agents based on bijective mappings, which can be instantiated with normalizing-flow neural networks. Secondly, we focus on the dual-robot case of $n = 2$ and present a translation-invariant controller that relies on a modified normalizing-flow neural network. The section is concluded by remarks on the proposed controllers.

A. Consensus-based Normalizing-Flow Controller

We give the following theorem for nonlinear consensus control of n robots:

Theorem 1: Let a system of $n \geq 2$ robotic agents evolving according to eq.(3) under a static and undirected communication graph $\mathscr{G}$. Further, let a differentiable and bijective mapping $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^d$, with $\mathbf{J} = \frac{\mathrm{d}\boldsymbol{\phi}(*)}{\mathrm{d}*} : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ being the respective Jacobian matrix, as well as constant, positive definite matrices $\mathbf{D}_i \in \mathbb{R}^{d\times d}$, $i \in \mathscr{N}$. If the graph is connected, the distributed control design

$$\boldsymbol{\tau}_i = \mathbf{g}_i(\mathbf{x}_i) - \mathbf{D}_i\dot{\mathbf{x}}_i - \mathbf{J}(\mathbf{x}_i)^\top \sum_{j\in\mathscr{N}_i} \big(\boldsymbol{\phi}(\mathbf{x}_i) - \boldsymbol{\phi}(\mathbf{x}_j)\big) \qquad (4)$$

guarantees that (i) $\lim_{t\to\infty}\big(\mathbf{x}_i(t) - \mathbf{x}_j(t)\big) = 0$, for all $i, j \in \mathscr{N}$, with $i \neq j$, if $\boldsymbol{\tau}_{\text{ext}} = 0$, and (ii) the closed-loop multi-agent system is passive if $\boldsymbol{\tau}_{\text{ext}} \neq 0$.

Proof: We first write the control design of eq. (4) in stacked vector form

$$\boldsymbol{\tau} = \mathbf{g}(\mathbf{x}) - \mathbf{D}\dot{\mathbf{x}} - \bar{\mathbf{J}}(\mathbf{x})^\top(\mathbf{L}\otimes I_n)\bar{\boldsymbol{\phi}}(\mathbf{x}), \qquad (5)$$

where $\bar{\boldsymbol{\phi}}(\mathbf{x}) = [\boldsymbol{\phi}(\mathbf{x}_1)^\top, \ldots, \boldsymbol{\phi}(\mathbf{x}_n)^\top]^\top \in \mathbb{R}^{nd}$, $\bar{\mathbf{J}}(x) = \mathrm{diag}\{\mathbf{J}(\mathbf{x}_1), \ldots, \mathbf{J}(\mathbf{x}_n)\} \in \mathbb{R}^{nd\times nd}$, and $\mathbf{D} = \mathrm{diag}\{\mathbf{D}_1, \ldots, \mathbf{D}_n\} \in \mathbb{R}^{nd\times nd}$. Consider now the continuously differentiable function

$$V = \frac{1}{2}\bar{\boldsymbol{\phi}}(\mathbf{x})^\top(\mathbf{L}\otimes I_n)\bar{\boldsymbol{\phi}}(\mathbf{x}) + \frac{1}{2}\dot{\mathbf{x}}^\top\mathbf{M}(\mathbf{x})\dot{\mathbf{x}}.$$

Since the multi-robot communication graph $\mathscr{G}$ is connected, $\mathbf{L}$ is positive semidefinite and hence V is non-negative.

We tackle first the case where $\boldsymbol{\tau}_{\text{ext}} = 0$. By differentiating V and using eq. (2) and the skew symmetry of $\dot{\mathbf{M}} - 2\mathbf{C}$, we obtain

$$\dot{V} = \dot{\mathbf{x}}^\top\big(\boldsymbol{\tau} - \mathbf{g}(\mathbf{x}) + \bar{\mathbf{J}}(\mathbf{x})^\top(\mathbf{L}\otimes I_n)\bar{\boldsymbol{\phi}}(\mathbf{x}) + \boldsymbol{\tau}_{\text{ext}}\big),$$

which, by substituting eq. (5), becomes $\dot{V} = -\dot{\mathbf{x}}^\top\mathbf{D}\dot{\mathbf{x}}$. Hence, $V(t)$ remains bounded, for all $t \geq 0$. Since $\boldsymbol{\phi}$ and $\mathbf{M}$ are a continuous, we also conclude the boundedness of the solution $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$, for all $t \geq 0$. Therefore, according to LaSalle's invariance principle [43, Chapter 4], the solution $\mathbf{x}(t)$ converges to the largest invariant set M in $E = \{(\mathbf{x}, \dot{\mathbf{x}}) \in \mathbb{R}^{2nd} : \dot{V} = 0\}$; E consists of all the points satisfying $\dot{\mathbf{x}} = 0$, due to the positive definiteness of $\mathbf{D}$. In view of the closed-loop system, consisting of eq. (2) and (5), the largest invariant set in E is the set $M = \{(\mathbf{x}, \dot{\mathbf{x}}) \in \mathbb{R}^{2nd} : \dot{\mathbf{x}} = 0, \ddot{\mathbf{x}} = 0\}$. Therefore, by substituting eq. (5) in eq. (2), we conclude that M consists of all the points that satisfy $\bar{\mathbf{J}}(\mathbf{x})^\top(\mathbf{L}\otimes I_n)\bar{\boldsymbol{\phi}}(\mathbf{x}) = 0$. Since $\boldsymbol{\phi}$ is a bijective mapping, $\bar{\mathbf{J}}^\top$ has full rank and, therefore, $\bar{\mathbf{J}}(\mathbf{x})^\top(\mathbf{L}\otimes I_n)\bar{\boldsymbol{\phi}}(\mathbf{x}) = 0$ implies $(\mathbf{L}\otimes I_n)\bar{\boldsymbol{\phi}}(\mathbf{x}) = 0$. Based on the properties of $\mathbf{L}$, we conclude that the system converges to the set M, where $\bar{\boldsymbol{\phi}}(\mathbf{x}) = \mathbf{1}\otimes\mathbf{c}$, for a constant $\mathbf{c} \in \mathbb{R}^d$ i.e., to the set where $\boldsymbol{\phi}(\mathbf{x}_1) = \boldsymbol{\phi}(\mathbf{x}_2) = \cdots = \boldsymbol{\phi}(\mathbf{x}_n)$. Since $\boldsymbol{\phi}$ is bijective, the latter implies $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n$, which proves part (i).

Next, we consider the case where $\boldsymbol{\tau}_{\text{ext}} \neq 0$. By following similar steps, the derivative of V becomes $\dot{V} = \dot{\mathbf{x}}^\top\boldsymbol{\tau}_{\text{ext}} - \dot{\mathbf{x}}^\top D\mathbf{x} \leq \dot{\mathbf{x}}^\top\boldsymbol{\tau}_{\text{ext}}$, which implies that the multi-agent system is passive under $\boldsymbol{\tau}_{\text{ext}}$, proving part (ii). ■

B. Translation-invariant Normalizing-Flow Controller

The controller presented in the prior section is not translation-invariant with respect to inter-agent differences $\mathbf{x}_i - \mathbf{x}_j$; that is, the control output does not remain identical when all robot positions are translated by a vector $\mathbf{c} \in \mathbb{R}^d$. Such a property can be useful for generalization of the controller to different setups, where the robotic agents are placed in another region of the workspace. In what follows, we propose a translation-invariant distributed controller for the special case of $n = 2$ robotic agents, using a modified normalizing-flow model.

Theorem 2: Let a system of $n = 2$ robotic agents evolving according to eq. (3) under a static and undirected communication graph $\mathscr{G}$. Further, let a differentiable and bijective mapping $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^d$, satisfying $\boldsymbol{\phi}(-*) = -\boldsymbol{\phi}(*)$, with $\mathbf{J} = \frac{\mathrm{d}\boldsymbol{\phi}(*)}{\mathrm{d}*} : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ being the respective Jacobian matrix, as well as positive definite matrices $\mathbf{D}_i, \mathbf{S} \in \mathbb{R}^{d\times d}$, $i \in \mathscr{N} = \{1, 2\}$. If the graph $\mathscr{G}$ is connected, the distributed control design

$$\begin{aligned}\boldsymbol{\tau}_i = \mathbf{g}_i(\mathbf{x}_i) - \mathbf{D}_i\dot{\mathbf{x}}_i - \mathbf{J}(\mathbf{x}_i - \mathbf{x}_{-i})^\top\mathbf{S}\boldsymbol{\phi}(\mathbf{x}_i - \mathbf{x}_{-i}) \\ + \mathbf{J}(\mathbf{x}_{-i} - \mathbf{x}_i)^\top\mathbf{S}\boldsymbol{\phi}(\mathbf{x}_{-i} - \mathbf{x}_i),\end{aligned} \qquad (6)$$

with $\mathbf{x}_{-1} = \mathbf{x}_2$, $\mathbf{x}_{-2} = \mathbf{x}_1$, guarantees that (i) $\lim_{t\to\infty}\big(\mathbf{x}_1(t) - \mathbf{x}_2(t)\big) = 0$, if $\boldsymbol{\tau}_{\text{ext}} = 0$ and (ii) the closed-loop multi-agent system is passive if $\boldsymbol{\tau}_{\text{ext}} \neq 0$.

Proof: We first write the control design eq. (6) in stacked vector form

$$\boldsymbol{\tau} = \mathbf{g}(\mathbf{x}) - \mathbf{D}\dot{\mathbf{x}} - \bar{\mathbf{L}}\bar{\mathbf{J}}(\bar{\mathbf{L}}\mathbf{x})^\top\bar{\mathbf{S}}\bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x}), \qquad (7)$$

where $\bar{\mathbf{L}} = \mathbf{L}\otimes I_n$, $\bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x}) = [\boldsymbol{\phi}(\mathbf{x}_1 - \mathbf{x}_2)^\top, \boldsymbol{\phi}(\mathbf{x}_2 - \mathbf{x}_1)^\top]^\top$, $\bar{\mathbf{J}}(\bar{\mathbf{L}}\mathbf{x}) = \mathrm{diag}\{\mathbf{J}(\mathbf{x}_1 - \mathbf{x}_2), \mathbf{J}(\mathbf{x}_2 - \mathbf{x}_1)\}$, $\mathbf{D} = \mathrm{diag}\{\mathbf{D}_1, \mathbf{D}_2\}$,

$\bar{\mathbf{S}} = \mathrm{diag}\{\mathbf{S},\mathbf{S}\}$, and in this 2-agent case, $\mathbf{L} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$. Consider now the continuously differentiable function

$$V = \frac{1}{2}\bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x})^\top \bar{\mathbf{S}}\bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x}) + \frac{1}{2}\dot{\mathbf{x}}^\top \mathbf{M}(\mathbf{x})\dot{\mathbf{x}},$$

which is non-negative due to the connectedness of $\mathscr{G}$ and hence the positive definiteness of $\mathbf{L}$. We tackle first the case where $\boldsymbol{\tau}_{\text{ext}} = 0$. By differentiating V and using eq. (3) and the skew-symmetry of $\dot{\mathbf{M}} - 2\mathbf{C}$, we obtain

$$\dot{V} = \bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x})^\top \bar{\mathbf{S}}\bar{\mathbf{J}}(\bar{\mathbf{L}}\mathbf{x})\bar{\mathbf{L}}\dot{\mathbf{x}} + \dot{\mathbf{x}}^\top(\boldsymbol{\tau} - \mathbf{g}(\mathbf{x})),$$

which, by substituting eq. (7), becomes $\dot{V} = -\dot{\mathbf{x}}^\top \mathbf{D}\dot{\mathbf{x}}$. Hence, $V(t)$ remains bounded, for all $t \geq 0$. Since $\boldsymbol{\phi}$ and $\mathbf{M}$ are continuous, we also conclude the boundedness of the solution $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$, for all $t \geq 0$. Therefore, according to LaSalle's invariance principle [43, Chapter 4], the solution $\mathbf{x}(t)$ converges to the largest invariant set M in $E = \{(\mathbf{x},\dot{\mathbf{x}}) \in \mathbb{R}^{2nd} : \dot{V} = 0\}$; E consists of all the points satisfying $\dot{\mathbf{x}} = 0$, due to the positive definiteness of $\mathbf{D}$. In view of the closed-loop system, consisting of eq. (2) and (5), the largest invariant set in E is the set $M = \{(\mathbf{x},\dot{\mathbf{x}}) \in \mathbb{R}^{2nd} : \dot{\mathbf{x}} = 0, \ddot{\mathbf{x}} = 0\}$. Therefore, by substituting eq. (7) in eq. (2), we conclude that M consists of all the points that satisfy $\bar{\mathbf{L}}\bar{\mathbf{J}}(\mathbf{L}\mathbf{x})^\top \bar{\mathbf{S}}\bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x}) = 0$, which reads

$$\mathbf{J}(\mathbf{x}_1 - \mathbf{x}_2)^\top \mathbf{S}\boldsymbol{\phi}(\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{J}(\mathbf{x}_2 - \mathbf{x}_1)^\top \mathbf{S}\boldsymbol{\phi}(\mathbf{x}_2 - \mathbf{x}_1) \quad (8)$$

Since $\boldsymbol{\phi}(-*) = -\boldsymbol{\phi}(*)$, it holds that $\mathbf{J}(-*) = \mathbf{J}(*)$ and $\boldsymbol{\phi}(0) = 0$. Furthermore, $\boldsymbol{\phi}$ is bijective and hence $\mathbf{J}^\top$ has full rank. By further using the positive definiteness of $\mathbf{S}$, eq. (8) becomes $\boldsymbol{\phi}(\mathbf{x}_1 - \mathbf{x}_2) = 0$. Since $\boldsymbol{\phi}$ is bijective, the latter implies that $\mathbf{x}_1 = \mathbf{x}_2$, which proves part (i).

Next, we consider the case where $\boldsymbol{\tau}_{\text{ext}} \neq 0$. By following similar steps, the derivative of V becomes $\dot{V} = \dot{\mathbf{x}}^\top \boldsymbol{\tau}_{\text{ext}} - \dot{\mathbf{x}}^\top \mathbf{D}\dot{\mathbf{x}} \leq \dot{\mathbf{x}}^\top \boldsymbol{\tau}_{\text{ext}}$, which implies that the multi-robot system is passive under $\boldsymbol{\tau}_{\text{ext}}$, proving part (ii). ■

C. Antipodal-Equivariant Normalizing-Flow Networks

The translation-invariant controller of Section IV-B relies on the property $\boldsymbol{\phi}(-*) = -\boldsymbol{\phi}(*)$. This effectively requires a normalizing-flow model that is equivariant under the reflection action at the origin, a.k.a. antipodal-equivariant. A closer look into the RealNVP layers eq. (1) reveals that we can achieve this by crafting neural networks $\mathbf{s}(*)$ and $\mathbf{t}(*)$ that are respectively invariant and equivariant under this action. To this end, we propose a simple modification without introducing extra architectures and parameters, by constructing even and odd functions:

$$\widetilde{\mathbf{s}}_k(*) = \mathbf{s}_k(*) + \mathbf{s}_k(-*)$$
$$\widetilde{\mathbf{t}}_k(*) = \mathbf{t}_k(*) - \mathbf{t}_k(-*)$$

We can then obtain an antipodal-equivariant bijection $\boldsymbol{\phi}(*)$ since function composition reserves equivariance property.

D. Remarks

The proposed algorithms guarantee the consensus of the agents' generalized coordinates $\mathbf{x}_i$; for articulated dynamics, these may correspond to joint-space variables, or Cartesian-space position coordinates. For redundant agents, the position coordinate is not a generalized coordinate and joints may be subject to nullspace motions. As discussed in the Chapter 8 and 9 of [44], Cartesian space control can be implemented by Jacobian transpose with nullspace motion accounted by joint space kinetic energy. Also, joint motion is commonly damped by viscous frictions therefore nullspace motions tend to stop at an equilibrium position.

We can extend the proposed consensus algorithms to account for first-order dynamics $\dot{x}_i = \tau_i,\ i \in \mathscr{N}$. More specifically, the controllers of eq. (5) and eq. (7) become $\boldsymbol{\tau} = \dot{\mathbf{x}} = -\bar{\mathbf{J}}(\mathbf{x})^{-1}\bar{\boldsymbol{\phi}}(\mathbf{x})$ and $\boldsymbol{\tau} = \dot{\mathbf{x}} = -\bar{\mathbf{J}}(\bar{\mathbf{L}}\mathbf{x})^{-1}\bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x})$, respectively, with the notation $\bar{\boldsymbol{\phi}}(\mathbf{x}) = [\boldsymbol{\phi}(\mathbf{x}_1)^\top, \ldots, \boldsymbol{\phi}(\mathbf{x}_n)^\top]^\top$, $\bar{\mathbf{J}}(x) = \mathrm{diag}\{\mathbf{J}(\mathbf{x}_1), \ldots, \mathbf{J}(\mathbf{x}_n)\}$, $\bar{\mathbf{L}} = \mathbf{L} \otimes I_n$, $\bar{\boldsymbol{\phi}}(\bar{\mathbf{L}}\mathbf{x}) = [\boldsymbol{\phi}(\mathbf{x}_1 - \mathbf{x}_2)^\top, \boldsymbol{\phi}(\mathbf{x}_2 - \mathbf{x}_1)^\top]^\top$, $\bar{\mathbf{J}}(\bar{\mathbf{L}}\mathbf{x}) = \mathrm{diag}\{\mathbf{J}(\mathbf{x}_1 - \mathbf{x}_2), \mathbf{J}(\mathbf{x}_2 - \mathbf{x}_1)\}$. We omit the detailed proofs since they are similar to the ones of Theorems 1 and 2.

It is worth noting that, although the second controller (eq. (6)) is translation-invariant, it does not necessarily generate equivariant position trajectories. This is attributed to the inertia and Coriolis terms $\mathbf{M}_i(\mathbf{x}_i)$, $\mathbf{C}_i(\mathbf{x}_i, \dot{\mathbf{x}}_i)$, which appear in the closed-loop system since the controller does not cancel them. Such terms vary across the spatial space and hence alter the eventual acceleration of the two robots when these are translated. The accommodation of equivariant trajectories may be achieved from online adaptation and inverse dynamics control algorithms, which we leave for the future work, or use the kinematics controller in the prior paragraph if tasks are less demanding on force control. Still, for motions that are not highly dynamic, the controller may still generalize to some extent as we demonstrate in the experiment section.

Both controllers have a nonlinear form constructed by normalizing-flow neural networks. Replacing them with an ordinary neural network implies the possibility of having a singular Jacobian, which will invalidate the zero point discussion e.g. in eq. (8). From our observations on numerical results, the normalizing-flows appear both capable of generating diverse trajectories. The main difference between the two forms, besides the number of robotic agents and the normalizing-flow networks, is whether the nonlinearity is applied before or after the multiplication of Laplacian. We find that the first controller appears more susceptible to numerical instability in integration, especially when a larger normalizing-flow model is used. This can be attributed to the fact that the first controller performs subtractions in the transformed space which may result in excessively large stiffness forces when the space is highly warped.

V. EXPERIMENTAL RESULTS

This section validates the proposed algorithms with some numerical examples and a case study on simulated dual-arm assembly in the context of reinforcement learning.

A. Normalizing-Flow-based Consensus

We consider numerical examples of point mass agents that validate the convergence of the proposed controllers. For the sake of visualization, we stick to planar motion by setting $d = 2$ and we test five simulation instances. The results are depicted in Figs. 4 and 5. More specifically, Fig. 4 shows the time evolution of the planar position trajectories of $n = 4$ robotic agents with mass $m_i = 0.1$, $\mathbf{D}_i = \mathbf{I}_{2\times 2}$, $i \in 1,2$, under the first control algorithm (see eq. (4)) where we select a random $\boldsymbol{\theta}$ for a two-layer $\boldsymbol{\phi}(*)$. All trajectories start with randomly sampled positions and velocities and converge at different points. The trajectories also show diverse transient behaviors thanks to the control nonlinearity. Fig. 5a depicts the generated position paths in the planar space under the second control algorithm (eq. (6)) for $n = 2$ robotic agents with mass $m_i = 0.5$, $\mathbf{D}_i = \mathbf{I}_{2\times 2}$, and a random equivariant $\boldsymbol{\phi}(*)$ of six RealNVP layers. One can verify that the agents successfully reach consensus. To validate the translation-invariance property of the proposed algorithm, we simulate the system from different initial conditions that retain the relative position of the two agents. As illustrated in Fig. 5b, the shape of the paths is reserved under the translation.

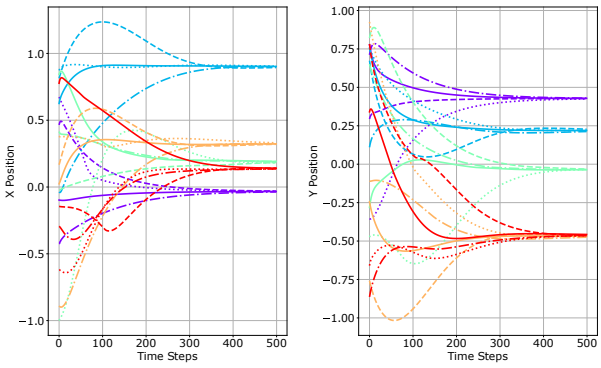
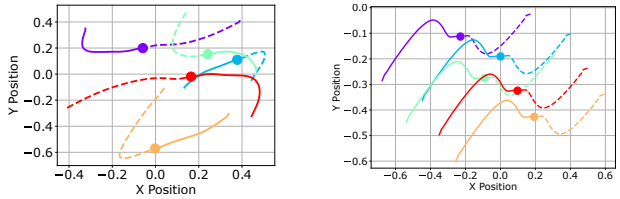


Fig. 4: Evolution of the planar position trajectories of $n = 4$ point-mass robotic agents under the distributed consensus-based normalizing-flow control algorithm (4). Simulation instances and agents are marked by different line colors and styles respectively.

B. Reinforcement Learning with Normalizing-Flow Policies

In this study, we show that the proposed controllers can be utilized for efficient learning of cooperative robotic manipulation skills. We consider learning the control parameters through reinforcement learning (RL) in an environment resembling a dual-arm assembly task, as is



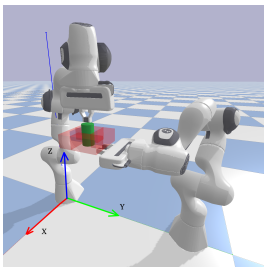
(a) Random initial states. (b) Initial states retaining relative position.

Fig. 5: Planar position paths of two point-masses robotic agents under the translation-invariant control algorithm eq.(6): (a) Paths from random initial positions (colored lines); (b) Paths from initial conditions retaining the relative positions of the agents. Simulation instances and agents are marked by different line colors and styles respectively.

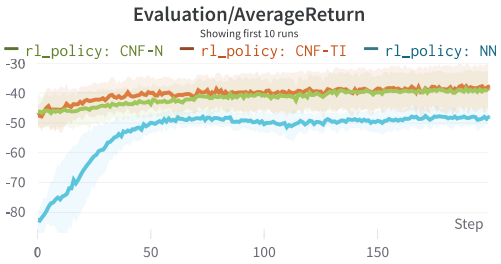
shown in Fig. 6a. The goal of the task is to insert a peg into a socket with a clearance of 2mm. Each part is rigidly attached to a torque-controlled Franka robot. The RL policy observes 3D translational position and velocity of the bottom centers of the objects, and outputs force actuation at these two points in the Cartesian space. The actuation is realized in the joint space through Jacobian transpose. The control also augments gravity compensation for attached objects and robots. An extra stiffness controller is used to regulate end-effector orientations. Due to the rotational compliance, the object positions are not strictly following the generalized coordinate dynamics assumed in the theorems. The policies thus need to learn to overcome this mismatch and to skillfully accommodate contacts while aligning the two parts.

The task environment is episodic with a horizon of $T = 250$ time steps. At the beginning of each episode, the end-effectors are randomly initialized in two 10cm$\times$10cm areas in the XY plane at fixed heights. A reward is assigned at each step based on the distance between the observed positions. We also add force magnitude to penalize large control efforts and evaluate the object distance with a scale of 10 at the terminal step. The insertion is deemed as a success when the final distance is below a threshold. The task is implemented at a simulated environment based on the physics engine PyBullet [45].

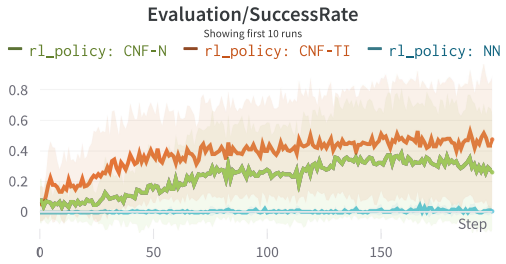
We use Proximal Policy Optimization (PPO) [46] and its implementation in the Garage library [47] for RL training. Normalizing-flow controllers are used to construct stochastic policies with addictive Gaussian noises. In particular, we set the $N = 2$ to the first controller with a name of CNF-N policy and label the policy using the translation-invariant one as CNF-TI. We compare learning performance of these policies to a fully-connected neural network baseline. All policies share same PPO parameters, value functions and neural networks of a similar size. The policies are trained through 200 epochs with each collecting 20 episodes for learning and evaluation. This amounts to 10^6 environment steps in total. We would like to clarify that although the CNF



(a) Dual-arm insertion environment



(b) Average return of evaluated episodes



(c) Success rate of evaluated episodes

Fig. 6: Reinforcement learning for developing a coordination strategy in a dual-arm assembly task. The policies that embed consensus-based normalizing-flow controllers outperform a deep neural network baseline with a significant margin.

policies can be executed in a distributed manner the training is still centralized under the standard PPO.

Fig. 6b reports RL learning curves averaged by 10 random seeds. The CNF policies benefit from the convergence property, showing a much better initial performance and constant improvement. The standard neural networks, the NN policy, also learns to bring the two parts closer from the shaped reward but performs consistently worse than CNF and cannot learn a more delicate coordination to complete the insertion. The gap is more evident from Fig. 6c which shows the success rates throughout the training process. The CNF policies starts managing successful insertions in the very beginning of the training stage. Both CNF-N and CNF-TI fail or unlearn on a few seeds while shows steady learning on the remained ones and can reach a more than 90% success rate. In contrast, the NN baseline barely succeeds with the allocated training budget. The comparison highlights some significant challenges posed to existing RL approaches in learning to coordinate two arms in a force control task. The proposed consensus-based normalizing-flow controllers can be used as efficient structure to alleviate the challenges.

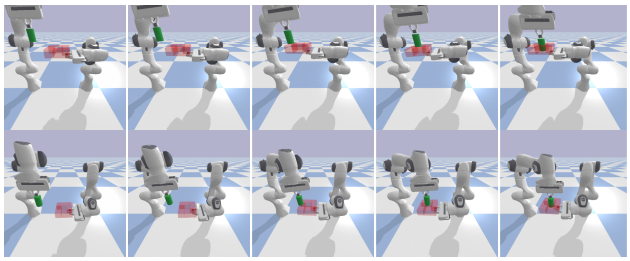


Fig. 7: Snapshots of CNF-TI generalized insertions from novel initial positions. Upper: +20cm; Bottom: -20cm.

As discussed in Section IV-D, the translation-invariant controller cannot guarantee to reserve the trajectory shape under general manipulator dynamics. Still, we are interested to see to what extent the learned policy can generalize to novel positions. We evaluate the best trained CNF policies in environments by translating initial end-effector positions along the Z axis, which are never trained in the RL process. Table I shows the number of successful insertions at positions of different offset. Our finding is that, even though the mass in Cartesian space is not a constant, the CNF-TI policy still manages to succeed most of the time when the testing positions are not far from the training area, see Fig. 7 for the snapshot of motion execution. The policy even generalize perfectly in the upward direction and it may generalize even further because the offset of 0.3m reaches the workspace limit. The negative direction appears more challenging to CNF-TI. We observe that typical failures are caused by unexpected collisions before sliding on the block surface. The policy can usually recover by bringing the peg back to the surface but cannot finish the insertion in time.

	offset (m)	0.05	0.10	0.15	0.20	0.25	0.30
CNF-N	+z axis	5/5	3/5	1/5	0/5	0/5	0/5
	-z axis	4/5	4/5	4/5	3/5	2/5	2/5
CNF-TI	+z axis	5/5	5/5	5/5	5/5	5/5	5/5
	-z axis	4/5	4/5	2/5	4/5	4/5	1/5

TABLE I: Number of successful insertions in 5 trials from untrained initial positions. The offsets indicate the vertical displacement of both end-effectors.

VI. CONCLUSIONS

This paper addresses learning skillful and dexterous dual-arm motion for complex manipulation tasks. We propose control algorithms that feature a free parameter space for generating rich trajectories with consensus stabilization. The richness comes from nonlinearity of diffeomorphic transformation, which are implemented by normalizing-flow networks and modified models of an equivariant form. Integrating with reinforcement learning, the proposed algorithms show a significant performance boost in a dual-arm manipulation setting. The robots manage to acquiring complex maneuvering skills to avoid unnecessary collisions and accommodate interaction forces in aligning two assembly parts. This task is shown to be particularly challenging and fail learning with a standard deep neural network policy. The learned skills are also demonstrated to generalize under significant end-effector displacement, highlighting

the importance of imposed consensus and equivariance structure.

For the outlook of future work, we plan to extend the translation-invariant controller to $N \geq 2$ for general multi-robot consensus problems. Diffeomoprhic transformations could be used as an effective framework to extend results under a linear assumption to a nonlinear form. It would also be interesting to investigate protocols for a dynamical Laplacian and convergence to other equilibrium configurations. For learning dual-arm motions, research efforts are needed to explore the representation capacity of these stable controllers. The controllers might also be useful for modeling human bimanual data and draw inspirations from them to encode other motion patterns. In addition to controlling two manipulators, the proposed algorithms may also benefit tasks entailing coordination of multiple end-effectors such as dexterous manipulation of multi-fingered hands.

ACKNOWLEDGMENT

The authors would like to show their gratitude to the European Research Council, Swedish Research Council and Knut and Alice Wallenberg Foundation.

References

[1] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation—a survey," Robotics and Autonomous Systems, vol. 60, no. 10, pp. 1340–1353, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901200108X

[2] K. Yao, D. Sternad, and A. Billard, "Hand pose selection in a bimanual fine-manipulation task," Journal of Neurophysiology, vol. 126, no. 1, pp. 195–212, 2021. [Online]. Available: http://infoscience.epfl.ch/record/287941

[3] S. Tarbouriech, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, and D. Sallé, "Dual-arm relative tasks performance using sparse kinematic control," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 6003–6009.

[4] J. Silvério, S. Calinon, L. Rozo, and D. G. Caldwell, "Learning task priorities from demonstrations," IEEE Transactions on Robotics, vol. 35, no. 1, pp. 78–94, 2019.

[5] V. Aladele and S. Hutchinson, "Impedance-based collision reaction strategy via internal stress loading in cooperative manipulation," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 5837–5843.

[6] C. K. Verginis and D. V. Dimarogonas, "Energy-optimal cooperative manipulation via provable internal-force regulation," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 9859–9865.

[7] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," Neural Computation, vol. 25, no. 2, pp. 328–373, 2013.

[8] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," IEEE Transactions on Robotics, vol. 27, no. 5, pp. 943–957, 2011.

[9] M. A. Rana, A. Li, D. Fox, B. Boots, F. Ramos, and N. Ratliff, "Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems," in Proc. Conference on Learning for Dynamics and Control (L4DC), 2020.

[10] S. A. Khader, H. Yin, P. Falco, and D. Kragic, "Learning stable normalizing-flow control for robotic manipulation," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 1644–1650.

[11] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in Proceedings of the 2005, American Control Conference, 2005., 2005, pp. 1859–1864 vol. 3.

[12] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," Proceedings of the IEEE, vol. 95, no. 1, pp. 215–233, 2007.

[13] M. Mesbahi and M. Egerstedt, Graph theoretic methods in multiagent networks. Princeton University Press, 2010.

[14] Z. Li, G. Wen, Z. Duan, and W. Ren, "Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs," IEEE Transactions on Automatic Control, vol. 60, no. 4, 2014.

[15] J. Sun, Z. Geng, and Y. Lv, "Adaptive output feedback consensus tracking for heterogeneous multi-agent systems with unknown dynamics under directed graphs," Systems & Control Letters, vol. 87, 2016.

[16] Z. Li, Z. Duan, and L. Huang, "Leader-follower consensus of multi-agent systems," in 2009 American control conference. IEEE, 2009, pp. 3256–3261.

[17] X. Zhang, L. Liu, and G. Feng, "Leader–follower consensus of time-varying nonlinear multi-agent systems," Automatica, vol. 52, pp. 8–14, 2015.

[18] Z. Li, X. Liu, P. Lin, and W. Ren, "Consensus of linear multi-agent systems with reduced-order observer-based protocols," Systems & Control Letters, vol. 60, no. 7, pp. 510–516, 2011.

[19] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," IEEE Transactions on Automatic Control, vol. 57, no. 5, pp. 1291–1297, 2011.

[20] C. Nowzari, E. Garcia, and J. Cortés, "Event-triggered communication and control of networked systems for multi-agent consensus," Automatica, vol. 105, pp. 1–27, 2019.

[21] C. P. Bechlioulis and G. A. Rovithakis, "Decentralized robust synchronization of unknown high order nonlinear multi-agent systems with prescribed transient and steady state performance," IEEE Transactions on Automatic Control, vol. 62, no. 1, pp. 123–134, 2016.

[22] S. A. Khader, H. Yin, P. Falco, and D. Kragic, "Stability-guaranteed reinforcement learning for contact-rich manipulation," IEEE Robotics and Automation Letters, vol. 6, no. 1, pp. 1–8, 2020.

[23] ——, "Learning deep energy shaping policies for stability-guaranteed manipulation," IEEE Robotics and Automation Letters, vol. 6, no. 4, 2021.

[24] K. Neumann and J. J. Steil, "Learning robot motions with stable dynamical systems under diffeomorphic transformations," Robotics and Autonomous Systems, vol. 70, pp. 1–15, 2015.

[25] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1530–1538.

[26] I. Kobyzev, S. Prince, and M. Brubaker, "Normalizing flows: An introduction and review of current methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[27] G. E. Henter, S. Alexanderson, and J. Beskow, "MoGlow: Probabilistic and controllable motion synthesis using normalising flows," ACM Transactions on Graphics, vol. 39, no. 4, pp. 236:1–236:14, 2020.

[28] W. Yin, H. Yin, D. Kragic, and M. Björkman, "Graph-based normalizing flow for human motion generation and reconstruction," in 2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN), 2021, pp. 641–648.

[29] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," The International Journal of Robotics Research, vol. 30, no. 3, pp. 324–334, 2011.

[30] S. Hirche et al., "Distributed control for cooperative manipulation with event-triggered communication," IEEE Transactions on Robotics, vol. 36, no. 4, 2020.

[31] C. K. Verginis, M. Mastellaro, and D. V. Dimarogonas, "Robust cooperative manipulation without force/torque measurements: Control design and experiments," IEEE Transactions on Control Systems Technology, vol. 28, no. 3, pp. 713–729, 2019.

[32] C. K. Verginis, W. S. Cortez, and D. V. Dimarogonas, "Adaptive cooperative manipulation with rolling contacts," in 2020 American Control Conference (ACC). IEEE, 2020, pp. 2735–2740.
[33] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings, vol. 1. IEEE, 2000, pp. 348–353.
[34] S. Erhart and S. Hirche, "Internal force analysis and load distribution for cooperative multi-robot manipulation," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1238–1243, 2015.
[35] C. K. Verginis, D. Zelazo, and D. V. Dimarogonas, "Cooperative manipulation via internal force regulation: A rigidity theory perspective," arXiv preprint arXiv:1911.01297, 2019.
[36] A. Z. Bais, S. Erhart, L. Zaccarian, and S. Hirche, "Dynamic load distribution in cooperative manipulation tasks," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 2380–2385.
[37] S. S. M. Salehian, N. Figueroa, and A. Billard, "A unified framework for coordinated multi-arm motion planning," The International Journal of Robotics Research, vol. 37, no. 10, pp. 1205–1232, 2018. [Online]. Available: https://doi.org/10.1177/0278364918765952
[38] D. Almeida and Y. Karayiannidis, "A lyapunov-based approach to exploit asymmetries in robotic dual-arm task resolution," in 2019 IEEE 58th Conference on Decision and Control (CDC), 2019.
[39] A. Sintov, S. Macenski, A. Borum, and T. Bretl, "Motion planning for dual-arm manipulation of elastic rods," IEEE Robotics and Automation Letters, 2020.
[40] M. Suomalainen, S. Calinon, E. Pignat, and V. Kyrki, "Improving dual-arm assembly by master-slave compliance," in 2019 International Conference on Robotics and Automation (ICRA), 2019.
[41] J. Wang, Z. Liu, and X. Hu, "Consensus of high order linear multi-agent systems using output error feedback," in Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference. IEEE, 2009, pp. 3685–3690.
[42] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," in Proceedings of International Conference on Learning Representations (ICLR), 2017.
[43] H. K. Khalil, Nonlinear systems; 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
[44] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control, 1st ed. Springer, 2008.
[45] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.
[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," CoRR, vol. abs/1707.06347, 2017.
[47] T. garage contributors, "Garage: A toolkit for reproducible reinforcement learning research," https://github.com/rlworkgroup/garage, 2019.