2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# Deep Geometric Potential Functions for Tracking on Manifolds

Nikhil Potu Surya Prakash[1], Joohwan Seo[1], Koushil Sreenath[1], Jongeun Choi[2] and Roberto Horowitz[1]

*Abstract*— **In this paper, we introduce a novel approach for designing invariant control laws through potential functions for fully actuated dynamical systems evolving on manifolds by leveraging the power of neural networks. The geometry and non-linearity inherent to manifold-based dynamical systems pose challenges for traditional control law design, necessitating techniques with the interplay of differential geometry and dynamical systems for ensuring stability. Apart from stability, performance and optimality are other challenging areas to address for dynamical systems evolving on manifolds. On top of these, the concept of invariance helps us improve learning transferability skills from one scene to another scene. We propose invariant potential functions on manifolds defined by neural networks that can be used to generate elastic forces for asymptotic tracking of trajectories. The weights of the potential function can be tuned to shape the potential functions according to the performance requirements through minimizing a loss function.**

## I. Introduction

The design of learning-based control laws for systems evolving on smooth manifolds remains an interesting problem in the field of control systems engineering. Such systems are important in a wide range of applications, like aerial robots [1], [2], [3], [4], [5], [6], robotic manipulators with impedance and admittance control systems [7], [8], [9], [10]. These applications demand sophisticated control strategies that can handle the inherent nonlinearities and complexities of the systems. In this context, the use of potential functions whose elastic forces dictate the control laws emerges as a viable controller synthesis approach [11], [12], offering a consistent way to guide system behavior towards desired states or trajectories.

It has been remarked that potential functions designed in [12], [1] on the Special Orthogonal group (SO(3)) suffer from vanishing gradients when the error between the desired and current orientation is $\pi$ radians. Consequently, due to the shape of the potential functions, at certain errors, the forces are small and have sluggish responses even when the errors are large. Research in [11], [13], [14] utilize a metric on the Lie algebras of SO(3) and Special Euclidean group (SE(3)) that is uniformly quadratic but the logarithmic map is not defined when the desired and current configurations are $\pi$ radians apart, since a rotation about any axis by $\pi$ radians would produce the same orientation. These potential functions can incorporate matrix gains for positions and orientations, which can be used to shape the response of the system, but it is not so straightforward to tune these gains easily based on the required response like in the case of linear systems. These shortcomings motivate us to use neural networks, for their representative power, to build potential functions whose gradients generate a part of the control law equivalent to the spring term/proportional term. We choose to build potential functions using neural networks instead of other basis functions as choosing basis functions that are expressive and representative require a lot of expertise. Whereas with neural networks, by simply increasing the depth and widths, expressiveness can be improved. These neural network potential functions might still have only almost global properties instead of global properties as it is not possible to construct a continuous control law that provides a continuous vector field on a compact manifold with a globally asymptotic equilibrium point [15]. Since we cannot overcome this, instead of having potential functions that have fixed properties, we wish to construct potential functions whose "problematic" points are made to appear in the regions on the manifold that we are not interested in.

Another important application of building potential functions using the representative power of neural networks is in learning potential fields via demonstrations by an expert in an invariant manner. In our previous work in [8], we provide a neural network architecture for learning gains as a function of the state from expert demonstrations using potential fields. The work in this paper can be used as a generalized framework for learning these state-dependent gains through potential functions, which provide a provably stabilizing control law directly during inference.

One of the challenges in building these potential functions is that their structure needs to satisfy specific Lyapunov function like properties, such as being zero at the equilibrium and positive everywhere else (or equivalently being lower bounded and attaining the minimum at its equilibrium). These properties will be achieved with the use of Input Convex Neural Networks (ICNNs) [16] and its application to the construction of Lyapunov functions to learn stable dynamical systems in [17]. Though these Lyapunov functions were constructed for Euclidean spaces, we will show in the next sections how this approach is also beneficial for systems evolving on smooth manifolds. The main application of the method will be for the orientation control of a spacecraft on SO(3) and for the Variable Impedance control of a robotic manipulator whose end effector pose evolves on SE(3). However, we will first show the necessary tools first for $\mathcal{S}^2$ since its geometry and properties are easy to visualize.
The main contributions of this paper can be summarized

[1]Department of Mechanical Engineering, University of California, Berkeley {nikhilps, joohwan_seo, koushils, horowitz}@berkeley.edu
[2]School of Mechanical Engineering, Yonsei University jongeunchoi@yonsei.ac.kr. Prof. Choi's work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF) (No.RS-2024-00344732)

as: 1. A detailed description of errors and variations on manifolds is presented. 2. A Neural Network architecture for building deep invariant geometric potential functions, which are consistent with the transport maps on smooth manifolds, whose variations can be defined, is developed. The potential functions generate stabilizing forces/wrenches for any random initialization of the network. 3. Kinematic control laws utilizing these potential functions and their stability are presented. 4. A Variable Impedance control law for a robotic manipulator and Orientation control law for a satellite using these potential functions are designed and a training procedure to shape the potential function for improving convergence is presented. An extension of this work to build dissipative potential functions along with conservative potential functions and for learning stabilizing controllers by mimicking human demonstrations has been presented in [18], [19].

## II. Preliminaries

In this section, we revisit the notion of an error on the manifold and describe the kinematics of a particle moving on the manifold, variations of configurations, and errors in velocities for the unit two sphere $\mathcal{S}^2$, the Special Orthogonal Group (SO(3)) and the Special Euclidean Group (SE(3)) from [12], [20].

### A. Two Sphere ($\mathcal{S}^2$):

A unit two sphere can be represented as follows

$$\mathcal{S}^2 = \{r \in \mathbb{R}^3 \ : \ \|r\|_2 = 1\}, \tag{1}$$

where $r$ is the coordinate of a point on the sphere with respect to a spatial frame attached to the origin (center of the sphere). The kinematics of a point whose configuration evolves on $\mathcal{S}^2$ can be written as

$$\dot{r} = \omega \times r = \hat{\omega} r, \tag{2}$$

where $\omega \in \mathbb{R}^3$ is an angular velocity vector and $\times$ represent the cross product. The hat $\hat{(\cdot)}$ and vee $(\cdot)^\vee$ maps are defined as follows for a vector $\eta = [\eta_1 \ \eta_2 \ \eta_3]^T \in \mathbb{R}^3$.

$$\hat{\eta} = \begin{bmatrix} 0 & -\eta_3 & \eta_2 \\ \eta_3 & 0 & -\eta_1 \\ -\eta_2 & \eta_1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & -\eta_3 & \eta_2 \\ \eta_3 & 0 & -\eta_1 \\ -\eta_2 & \eta_1 & 0 \end{bmatrix}^\vee = \eta. \tag{3}$$

The time derivative $\dot{r}$ is an element on the tangent space ($T_r\mathcal{S}^2$) of $\mathcal{S}^2$ at $r$. For a desired configuration $r_d \in \mathcal{S}^2$, the configuration error $r_e \in \mathbb{R}$ can be defined as

$$r_e \triangleq r_d^T r. \tag{4}$$

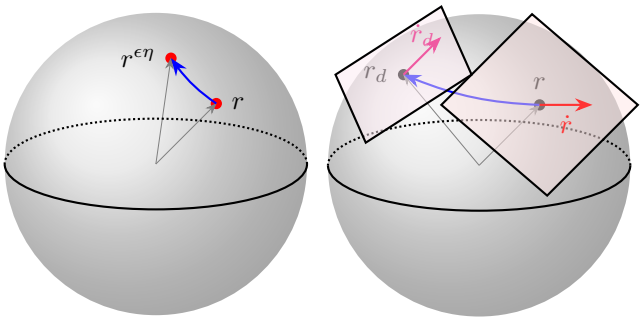
With the current definition of error, the control law should be designed to drive the error to 1 instead of 0.

A variation on the sphere can be defined as a configuration on the sphere obtained by flowing with an angular velocity $\eta \in \mathbb{R}^3$ for a time $\epsilon \in \mathbb{R}$ from the configuration $r$. Using the exponential map (see [21]), the variation $r_{\epsilon\eta}$ can be defined as follows

$$r_{\epsilon\eta} \triangleq \exp(\epsilon\hat{\eta}) r. \tag{5}$$

The infinitesimal variation $\delta r_\eta \in T_r\mathcal{S}^2$, can now be defined as

$$\delta r_\eta \triangleq \tfrac{d}{d\epsilon} r_{\epsilon\eta}\big|_{\epsilon=0} = \hat{\eta} r = \eta \times r. \tag{6}$$




(a) (b)

Fig. 1: (a) Variation of an element on $\mathcal{S}^2$ (b) Configurations and their Tangent spaces

The velocity error on $\mathcal{S}^2$ can now be defined using a transport map [12] $\mathcal{T} : T_{r_d}\mathcal{S}^2 \mapsto T_r\mathcal{S}^2$ as

$$e_v \triangleq \dot{r} - \mathcal{T}\dot{r}_d, \tag{7}$$

where the transport map $\mathcal{T}$ on $\mathcal{S}^2$ is given by

$$\mathcal{T} = (r_d^T r)\mathcal{I}_3 + \widehat{(r_d \times r)}. \tag{8}$$

Since the current velocity $\dot{r}$ and the desired velocity $\dot{r}_d$ Lie on different tangent spaces, i.e., $T_r\mathcal{S}^2$ and $\mathcal{T}_{r_d}\mathcal{S}^2$ respectively, they cannot be directly compared like in the case of Euclidean space. Therefore, it is necessary to transport one vector to the tangent space of the other and compare them. This comparison can be made with the help of this transport map [12], and the potential functions must be designed such that they are compatible with the transport map.

### B. Special Orthogonal Group ($SO(3)$):

The Special Orthogonal group (SO(3)) represents the set of all possible rotation matrices $R$ without any reflections. The following is a representation of the group as an embedding in $\mathbb{R}^{3\times3}$

$$SO(3) = \{R \in \mathbb{R}^{3\times3} \ : \ R^T R = \mathcal{I}_3 \ , det(R) = 1\}. \tag{9}$$

The kinematics of a body that just is restricted to rotate without any translations can be written as

$$\dot{R} = R\hat{\Omega}, \tag{10}$$

where $\Omega \in \mathbb{R}^3$ is the angular velocity expressed in the body-fixed frame and $\hat{\Omega} \in \mathfrak{so}(3)$, the Lie Algebra of SO(3) (tangent space at identity).

A variation on SO(3) consistent with the kinematics in (10) can be defined as

$$R_{\epsilon\eta} \triangleq R\exp(\epsilon\hat{\eta}). \tag{11}$$

The infinitesimal variation $\delta R_\eta \in T_R SO(3)$ can now be defined as

$$\delta R_\eta \triangleq \tfrac{d}{d\epsilon} R_{\epsilon\eta}\big|_{\epsilon=0} = R\hat{\eta}. \tag{12}$$

The configuration error between a desired configuration $R_d$ and the current configuration $R$ can be defined as

$$R_e \triangleq R_d^T R. \tag{13}$$

This error is called the right error according to [12]. Note that the error $R_e$ is also an element of $SO(3)$, and the error becomes $\mathcal{I}_3$ when $R = R_d$. The body fixed angular velocity error according to [12] can be defined using the body fixed

angular velocity $\hat{\Omega} = R^T\dot{R}$ and the desired angular velocity $\hat{\Omega}_d = R_d^T\dot{R}_d$ via

$$\begin{aligned}\dot{R}_e &= \tfrac{d}{dt}(R_d^T R) = \dot{R}_d^T R + R_d^T\dot{R}\\ &= \hat{\Omega}_d^T R_d^T R + R_d^T R\hat{\Omega} = -\hat{\Omega}_d R_e + R_e\hat{\Omega} \qquad (14)\\ &= R_e(\hat{\Omega} - R_e^T\hat{\Omega}_d R_e) = R_e\hat{\Omega}_e.\end{aligned}$$

Using $(R_e^T\hat{\Omega}_d R_e)^\vee = R_e^T\Omega_d$, we define the body fixed angular velocity error as

$$\Omega_e \triangleq \Omega - R_e^T\Omega_d. \qquad (15)$$

### C. Special Euclidean Group ($SE(3)$):

The Special Euclidean group (SE(3)) describes the pose of a rigid body in 3D space via a rotation matrix $R$ and a position $p$. The following is a representation of the group:

$SE(3) = \{(R,p) \in SO(3)\times\mathbb{R}^3 \ : \ R^TR = \mathcal{I}_3 \ , det(R) = 1\}$.

The kinematics of a body evolving on SE(3) can be written as

$$\dot{g} = g\Gamma(V^b), \qquad (16)$$

where

$$g = \begin{bmatrix} R & p\\ 0 & 1\end{bmatrix}, V^b = \begin{bmatrix} v\\ \Omega\end{bmatrix}, \Gamma(V^b) = \begin{bmatrix}\hat{\Omega} & v\\ 0 & 0\end{bmatrix}. \qquad (17)$$

Here $g$ is called the homogeneous representation of the SE(3) group and $v \in \mathbb{R}^3$ is the translational velocity with $\dot{p} = v$ represented in the body coordinates.

A variation on SE(3) can be defined using the exponential map (see [21]) as

$$g_{\epsilon\eta} \triangleq g\exp(\epsilon\Gamma(\eta)) \qquad (18)$$

where $\eta = [\eta_1^T, \eta_2^T]^T$ for $\eta_1, \eta_2 \in \mathbb{R}^3$ and note that we regard $\eta_1$ as a translational element and $\eta_2$ as a rotational element following the convention of [21]. The infinitesimal variation can now be defined as

$$\delta g_\eta \triangleq \tfrac{d}{d\epsilon}g_{\epsilon\eta}\big|_{\epsilon=0} = g\Gamma(\eta) = \begin{bmatrix} R\hat{\eta}_2 & R\eta_1\\ 0 & 0\end{bmatrix}. \qquad (19)$$

The configuration error between a desired configuration $g_d$ and the current configuration $g$ can be defined as follows with $R_e$ from (13) and $p_e = p - p_d$ denoting the translational error

$$g_e \triangleq g_d^{-1}g = \begin{bmatrix} R_e & R_d^Tp_e\\ 0 & 1\end{bmatrix}, g_d = \begin{bmatrix} R_d & p_d\\ 0 & 1\end{bmatrix}. \qquad (20)$$

Note that the configuration error $g_e$ is also an element of $SE(3)$ and the error becomes $\mathcal{I}_4$ when $g = g_d$.

By taking the time derivative of $g_e$ and following the same steps as (14), we can obtain the following

$$\dot{g}_e = g_e\Gamma(e_V). \qquad (21)$$

where $e_V$ is the velocity error defined by the following utilizing the desired quantities with subscript $d$

$$e_V \triangleq \underbrace{\begin{bmatrix} v\\ \Omega\end{bmatrix}}_{V^b} - \underbrace{\begin{bmatrix} R_e^Tv_d + R_e^T\hat{\Omega}_d R_d^T(p-p_d)\\ R_e^T\Omega_d\end{bmatrix}}_{V_d^*} = \begin{bmatrix} e_v\\ e_\Omega\end{bmatrix}. \qquad (22)$$

*Left Invariance:*

It can be seen from the following equations that by transforming the current and desired configurations both from the left arbitrarily by the same translation, we do not get a change in the error.

$$\begin{aligned}&(R_lr_d)^T(R_lr) = r_d^T(R_l^TR_l)r = r_d^Tr = r_e\\ &(R_lR_d)^T(R_lR) = R_d^T(R_l^TR_l)R = R_d^TR = R_e \qquad (23)\\ &(g_lg_d)^{-1}(g_lg) = g_d^{-1}(g_l^{-1}g_l)g = g_d^{-1}g = g_e\end{aligned}$$

This is an essential property as incorporating this property allows us to transfer trained skills from one scene to another scene without any new training. We will use these errors to construct potentials in Sec. IV. Since the potentials depend solely on the errors, the potential functions are also left invariant. It is also easy to see that for left error representations, we have right invariance. [8] presents a more elaborate explanation of invariance.

*Variation of a function:*

For any scalar function $u : \mathcal{M} \mapsto \mathbb{R}$ mapping from a manifold $\mathcal{M}$ to the the reals, the derivative $D_mu$ at $m \in \mathcal{M}$ can be defined with the help of its infinitesimal variation $\delta u$ described as

$$\delta u(m) = \tfrac{d}{d\epsilon}u_{\epsilon\eta}\big|_{\epsilon=0} = \tfrac{d}{d\epsilon}u(m_{\epsilon\eta})\big|_{\epsilon=0} = D_mu(m)\cdot\eta, \qquad (24)$$

with $m_{\varepsilon\eta}$ as the variation of $m$ on $\mathcal{M}$ obtained by flowing from $m$ with $\eta$ for a period of $\epsilon$. Furthermore,

$$D_mu \triangleq \tfrac{\partial}{\partial\eta}\delta u \qquad (25)$$

This derivative $D_mu$ will be used to define the control law in the later sections.

## III. Standard Potential Functions

In this section, we will present two types of geometric potential functions that are common in the literature for $SO(3)$ manifolds. Though their simplicity makes them an easy choice for use in geometric control laws, a short discussion about their shortcomings will be presented.

### A. Potential Function 1:

First, we will consider the potential function from [12] utilizing the trace operator $tr[\cdot]$ which was later used in [1], [2], [3], [4], [22], [5], [23] etc. It was also shown that this potential function can be expressed using the Frobenius norm in [9].

$$\Psi_1(R_e) = tr[I - R_e] = tr[I - R_d^TR] \qquad (26)$$

$R, R_d, R_e = R_d^TR \in SO(3)$ and $\Psi_1 : SO(3) \mapsto \mathbb{R}^+$. The elastic force $f_1$ generated by this potential function can be obtained from its variation.

$$\begin{aligned}\delta\Psi_{1\,\eta} &= \tfrac{d}{d\epsilon}\Psi_{1,\epsilon\eta}\big|_{\epsilon=0} = \tfrac{d}{d\epsilon}tr[I - R_d^TR_{\epsilon\eta}]\big|_{\epsilon=0} \qquad (27)\\ &= -f_1\cdot\eta = (R_e - R_e^T)^\vee\cdot\eta\end{aligned}$$

A detailed derivation of how to obtain the forces will be presented in the next section.

### B. Potential Function 2:

We now consider the potentials (Lie algebra potentials) from [11], [13], [14] using the logarithmic map from the Lie group to its Lie algebra (see [21]) as follows

$$\Psi_2(R_e) = \tfrac{1}{2}\|log(R_e)\|_F^2 = \tfrac{1}{2}\|log(R_d^TR)\|_F^2 \qquad (28)$$

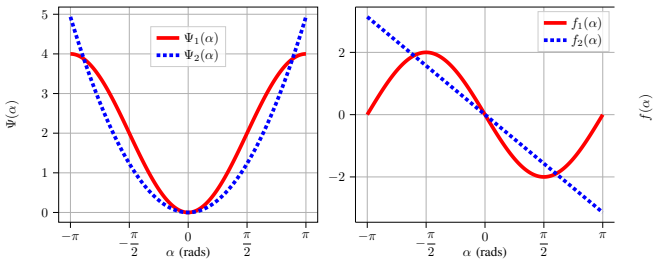



Fig. 2: Comparison of Potential Functions (left) and corresponding elastic forces (right).

where $\|A\|_F = tr[A^T A]^{1/2}$ represents the Frobenius norm of the matrix $A$. The elastic force $f_2$ generated from this potential can be obtained from its variation as follows

$$\begin{aligned}\delta\Psi_{2,\eta} &= \tfrac{d}{d\epsilon}\Psi_{2,\epsilon\eta}\big|_{\epsilon=0} = \tfrac{1}{2}\,\tfrac{d}{d\epsilon}\|log(R_d^T R_{\epsilon\eta})\|_F^2\big|_{\epsilon=0} \\ &= -f_2\cdot\eta = -log(R_e)^\vee\cdot\eta\end{aligned} \tag{29}$$

Fig. 2 shows a comparison between the potential function and their corresponding elastic forces for SO(2) manifold (planar rotations with $\alpha$ as the angle between desired and current configurations). Here we can see that for smaller errors, $\Psi_1$ is slightly better, and for larger errors, $\Psi_2$ outperforms as $f_1$ tends to become zero even when the error is $\pi$ radians. Since the rotations are planar, they can be viewed as rotations about a fixed axis and there is no ambiguity in the axis. But when it comes to 3D rotations, a rotation about any axis by $\pi$ radians causes the same orientation, which causes a discontinuity in $\Psi_2$, and therefore, the logarithmic map is defined only when the argument is not the identity. We wish to overcome these issues by using more expressive potential functions that can be built through neural networks.

## IV. Potential Functions

In this section, a methodology for designing positive definite potential functions utilizing Input Convex Neural Networks (ICNNs) [16] and its use in the construction of Lyapunov functions in [17] will be presented. We will take a similar approach to build the potential functions. But to accommodate matrix inputs, we will modify the structure for different manifolds, as shown in the upcoming subsections. The aim is to make the potential function zero when the current configuration and desired configuration coincide and positive everywhere else. Though convexity is a restriction, we can relax this by adding another layer of an invertible residual network [24] layer before ICNN as we need to have only one point where the elastic force becomes zero. A caveat here is that though we are using an ICNN to build the potential functions, the manifold is not convex, and hence, the potential function and domain pair together are not convex, but the potential function with a convex domain remains convex.

### A. Potential Functions on $\mathcal{S}^2$:

We can define the potential function $\Psi : [-1, 1] \mapsto \mathbb{R}^+$ such that $\Psi(1) = 0$ as follows:

$$\begin{aligned}&z_1 = \sigma_0(W_0 r_e + b_0)\\ &z_{i+1} = \sigma_i(U_i z_i + W_i r_e + b_i), i = 1,\dots,k-1\\ &\Phi(r_e) \equiv z_k\\ &\Psi(r_e) = \sigma_{k+1}(\Phi(r_e) - \Phi(1)) + \epsilon\,\|1 - r_e\|_2^2,\end{aligned} \tag{30}$$

where $i = 1, \cdots, k-1$. Here $z_i$ refers to the output of the $i^{th}$ layer. The weights $U_i \in \mathbb{R}^+$, $W_i \in \mathbb{R}$ and the bias $b_i \in \mathbb{R}$. The nonlinearities $\sigma_i$ are convex like ReLU. To ensure smooth gradients, the sharp edges of ReLU can be smoothed (see [17]). In the last equation in (30), we shift the function $\Phi$ to become zero when the error $r_e = 1$ and pass it through a non negative nonlinearity (like ReLU) $\sigma_{k+1}$ to make the function positive semi-definite. Finally through $\epsilon\,\|1 - r_e\|_2^2$ for $\epsilon > 0$, we make the entire function positive definite.

### B. Potential Functions on $SE(3)$

We can define $\Psi : SE(3) \mapsto \mathbb{R}^+$ such that $\Psi(\mathcal{I}_3, 0_{3\times1}) = 0$ as follows.

$$\begin{aligned}&z_1^l = \sigma_0(tr[W_0^l R_e] + \tilde{W}_0^l R_d^T p_e + b_0^l)\\ &z_{i+1}^l = \sigma_i(U_i^l z_i + tr[W_i^l R_e] + \tilde{W}_i^l R_d^T p_e + b_i^l),\\ &\Phi(R_e, R_d^T p_e) \equiv z_k\\ &\Psi(R_e, R_d^T p_e) = \sigma_{k+1}(\Phi(R_e, R_d^T p_e) - \Phi(\mathcal{I}_3, 0_{3\times1}))\\ &+\epsilon\,\|\mathcal{I}_4 - g_d^{-1} g\|_F^2,\end{aligned} \tag{31}$$

where $z_i^l$ refers to the output of $l^{th}$ neuron of $i^{th}$ layer. The weights $U_i^l \in \mathbb{R}^+$, $W_i^l \in \mathbb{R}^{3\times3}$, $\tilde{W}_i^l \in \mathbb{R}^3$ and bias $b_i^l \in \mathbb{R}$. Similar to (30), we shift the function $\Phi$ such that it is zero when $g_e = \mathcal{I}_4$ and make it positive definite by adding $\epsilon\,\|\mathcal{I}_4 - g_d^{-1} g\|_F^2$ for $\epsilon > 0$. By making $p_e = 0$, the potential function in (31) reduces to the potential on SO(3). It is worth noting here that the potential functions defined in (30) and (31) always produce a stabilizing elastic force irrespective of the initialization of the weights as long as $U$ matrices are non-negative. This constraint on $U$ can be enforced by first initializing them randomly and then using the softplus activation function to make them all positive.

## V. Kinematic Control

In this section, a detailed description of how to obtain the elastic forces from the potential functions will be presented for $\mathcal{S}^2$, SO(3) and SE(3) manifolds. Stability proofs for $\mathcal{S}^2$ and SO(3) will be presented as the extension to SE(3) from SO(3) is straightforward. Although we are more interested in applications to dynamics on manifolds, we will emphasize the derivations for kinematic systems as the control laws straightforwardly extend to dynamical systems by adding a damping term to the elastic force.

### A. Unit two sphere $\mathcal{S}^2$:

The variation of the potential function defined in (30) will be used to obtain the elastic force as follows

$$\begin{aligned}\Psi = \Psi(r_d^T r) \implies \Psi_{\epsilon\eta} &= \Psi(r_d^T e^{\epsilon\hat{\eta}} r)\\ \implies \delta\Psi_\eta = \tfrac{d}{d\epsilon}\Psi_{\epsilon\eta}\big|_{\epsilon=0} &= \Psi'(r_d^T r)(r_d^T e^{\epsilon\hat{\eta}}\hat{\eta} r)\\ &= \Psi'(r_d^T r)(r_d^T \hat{\eta} r)\end{aligned} \tag{32}$$

where $\Psi' \triangleq \frac{\partial}{\partial z}\Psi(z)\big|_{z=r_d^T r}$. Utilizing the cyclic property of the box product $[a\ b\ c] = a\cdot(b\times c) = a^T\hat{b}c$ that $[a\ b\ c] = [b\ c\ a] = [c\ a\ b]$, we can write

$$\delta\Psi = \Psi'(r_d^T r)(\hat{r}r_d)^T\eta = \Psi'(r_d^T r)(\hat{r}r_d)\cdot\eta = D_r\Psi\cdot\eta \tag{33}$$

The corresponding elastic force is now given by

$$f = -D_r\Psi = -\Psi'(r_d^T r)(\hat{r}r_d) \tag{34}$$

Since $\Psi$ is a neural network, its partials or gradients can be found using automatic differentiation packages like PyTorch's *autograd*.

**Theorem 1.** *The kinematic tracking control law $\omega = \omega_d + f$, with the elastic force $f$ given by $f = -\Psi'(r_d^T r)(\hat{r} r_d)$, almost globally asymptotically tracks $r_d(t)$ for the system* (2)*.*

*Proof.* We will consider the potential function $\Psi$ as the Lyapunov function candidate as it is positive definite, and the only point where it is 0 is when $r_e$ is 1. The time derivative of the Lyapunov function yields

$$\begin{aligned}\dot{\Psi} &= \Psi'(r_d^T r)(\dot{r}_d^T r + r_d^T \dot{r}) = \Psi'(r_d^T r)(r^T \hat{\omega}_d r_d + r_d^T \hat{\omega} r)\\ &= \Psi'(r_d^T r) r_d^T (\hat{\omega} - \hat{\omega}_d) r\\ &= \Psi'(r_d^T r) r_d^T (\hat{\omega}_d - \hat{\omega}_d - \Psi'(r_d^T r)(\widehat{r \times r_d})) r\end{aligned}$$

Utilizing the property that $a \times (b \times c) = b(a \cdot c) - c(a \cdot b)$

$$\begin{aligned}\dot{\Psi} &= -\Psi'(r_d^T r)^2 r_d^T (r_d \|r\|_2^2 - r(r_d \cdot r)) \qquad (35)\\ &= -\Psi'(r_d^T r)^2 (\|r_d\|_2^2 \|r\|_2^2 - (r_d \cdot r)^2) \leq 0.\end{aligned}$$

### *B. Special Orthogonal Group SO(3):*

A potential on SO(3) can be represented as $\Psi(R_e)$ by letting $p_e = 0$ in (31). The variation of this potential function defined will be used to obtain the elastic force as follows

$$\begin{aligned}&\Psi = \Psi(R_d^T R) \implies \Psi_{\epsilon\eta} = \Psi(R_d^T R e^{\epsilon\hat{\eta}})\\ \implies &\delta\Psi_\eta = \tfrac{d}{d\epsilon}\Psi_{\epsilon\eta}\big|_{\epsilon=0} = tr[\Psi'(R_d^T R)^T R_d^T R e^{\epsilon\hat{\eta}}\hat{\eta}] \qquad (36)\\ &\qquad = tr[\Psi'(R_d^T R)^T R_d^T R \hat{\eta}],\end{aligned}$$

where $\Psi' = \frac{\partial}{\partial M}\Psi(M)\big|_{M=R_e}$. Utilizing the property of trace that $-\frac{1}{2}tr[A\hat{b}] = (A^T - A)^\vee \cdot b$, we can write

$$\begin{aligned}&\delta\Psi_\eta = (R_e^T \Psi'(R_e) - \Psi'(R_e)^T R_e)^\vee \cdot \eta \qquad (37)\\ \implies &D_R\Psi = D_{R_e}\Psi = (R_e^T \Psi'(R_e) - \Psi'(R_e)^T R_e)^\vee.\end{aligned}$$

The corresponding elastic force is now given by

$$f = -D_R\Psi = -(R_e^T \Psi'(R_e) - \Psi'(R_e)^T R_e)^\vee. \qquad (38)$$

**Theorem 2.** *The kinematic tracking control law $\Omega = R_e^T \Omega_d + f$, where the elastic force $f$ is given by* (38)*, almost globally asymptotically tracks $R_d(t)$ for the system* (10)*.*

*Proof.* We will consider the potential function $\Psi$ as the Lyapunov function candidate as it is positive definite and the only point where it is 0 is when $R_e$ is $\mathcal{I}_3$. The time derivative of the Lyapunov function yields

$$\begin{aligned}\dot{\Psi} &= tr[\Psi'(R_e)^T \dot{R}_e] = tr[\Psi'(R_e)^T R_e \hat{\Omega}_e]\\ &= [(\Psi'^T R_e)^T - (\Psi'^T R_e)]^\vee \cdot \Omega_e.\end{aligned}$$

Utilizing the angular velocity error from (15), we get

$$\dot{\Psi} = -\|(R_e^T \Psi'(R_e) - \Psi'(R_e)^T R_e)^\vee\|_2^2 \leq 0. \qquad (39)$$

### *C. Special Euclidean Group SE(3):*

The variation of the potential function on SE(3) defined in (31) can be used to obtain the elastic force.

$$\begin{aligned}&\Psi = \Psi(R_d^T R, R_d^T p_e)\\ \implies &\Psi_{\epsilon\eta} = \Psi(R_d^T R e^{\epsilon\eta_2}, R_d^T(p + \epsilon R \eta_1 - p_d))\\ \implies &\delta\Psi_\eta = \tfrac{d}{d\epsilon}\Psi_{\epsilon\hat{\eta}}\big|_{\epsilon=0} = tr[\partial_1\Psi^T R_d^T R \hat{\eta}_2] + \partial_2\Psi^T R_d^T R \eta_1,\end{aligned}$$

where $\partial_1\Psi = \frac{\partial}{\partial M}\Psi(M,z)\big|_{M=R_e, z=R_d^T p_e}$ and $\partial_2\Psi = \frac{\partial}{\partial z}\Psi(M,z)\big|_{M=R_e, z=R_d^T p_e}$. This yields the elastic force as

$$f = \begin{bmatrix} -R_e^T \partial_2\Psi \\ -(R_e^T \partial_1\Psi - \partial_1\Psi^T R_e)^\vee \end{bmatrix}. \qquad (40)$$

**Theorem 3.** *The control law given by the following equation almost globally asymptotically tracks $g_d(t)$*

$$\begin{bmatrix}\Omega \\ v\end{bmatrix} = \begin{bmatrix} R_e^T \Omega_d \\ v_d \end{bmatrix} + f. \qquad (41)$$

*Proof.* For brevity, we skip the stability proof as the process is similar to that of SO(3) for rotations with a trivial extension to translations.

In every case, the time derivative of the Lyapunov function is zero only at a finite number of points when $f = 0$, but negative everywhere else, and therefore, the equilibrium is almost globally asymptotically stable. The remaining points are unstable equilibria.

## VI. Dynamic Control

In this section, we consider two interesting and practical problems on SO(3) and SE(3) manifolds namely orientation control of a satellite and Impedance control of a robotic manipulator respectively. The dynamics of both the systems and stabilizing control laws using the constructed potential functions will be presented.

### *A. Control of a Satellite on SO(3):*

A simple model of a rigid body rotating without translating can be used to describe the orientation control problem of a satellite. The orientation of the satellite is described through rotation matrices $R \in SO(3)$. The control is achieved through momentum wheels attached to three perpendicular axes of the satellite. Again for simplicity, we will ignore the dynamics of the reaction wheels and assume the availability of three independent torque components along its three perpendicular axes. The dynamics can be written as follows with $\mathbb{J}$ as the symmetric positive definite inertia matrix, $\Omega \in \mathbb{R}^3$ as the angular velocity represented in the body-fixed frame and $\tau \in \mathbb{R}^3$ as the torque.

$$\begin{aligned}\dot{R} &= R\hat{\Omega}\\ \mathbb{J}\dot{\Omega} + \hat{\Omega}\mathbb{J}\Omega &= \tau\end{aligned} \qquad (42)$$

**Theorem 4.** *The following control almost globally asymptotically tracks $R_d(t)$ for a dynamical system described by* (42) *with the elastic force $f$ described by* (38)

$$\tau = \hat{\Omega}\mathbb{J}\Omega - \mathbb{J}\hat{\Omega}_e R_e^T \Omega_d + \mathbb{J} R_e^T \dot{\Omega}_d + \mathbb{J}(f - K_D\Omega_e). \qquad (43)$$

*Proof.* This control law achieves the following error dynamics

$$\dot{\Omega}_e + K_D\Omega_e + D_R\Psi(R_e) = 0. \qquad (44)$$

We will consider the following positive definite Lyapunov candidate function

$$\begin{aligned}&W = \Psi(R_e) + \tfrac{1}{2}\Omega_e^T \Omega_e\\ \implies &\dot{W} = \dot{\Psi}(R_e) + \Omega_e^T \dot{\Omega}_e = \Omega_e^T(\dot{\Omega}_e - f)\\ &\qquad = -\Omega_e^T K_D \Omega_e \ \leq\ 0.\end{aligned}$$

Using Lasalle's Invariance principle, we can also conclude that the equilibrium $R_e = \mathcal{I}_3$ of the error dynamics in (44) is almost globally asymptotically stable as the largest invariant set where $\dot{W} = 0$ is only when $R_e = \mathcal{I}_3$ (removing the other unstable equilibria). It may firstly be a bit non-intuitive, but it should be noted that $\Omega_e$ can be expressed in terms of $R_e$ and $\dot{R}_e$ which makes the entire error equation a function of just $R_e$ and its time derivatives. The expression is omitted for the compactness of notation.

### *B. Control of a Robotic Manipulator on SE(3):*

Another important problem where potential functions play an important role is in the control of robotic manipulators. We will demonstrate an application to Impedance control of a robotic manipulator [9].

In the field of impedance control, it is well known from [25] that the robotic manipulator's dynamics can be rewritten as

$$\tilde{M}(q)\dot{V}^b + \tilde{C}(q,\dot{q})V^b + \tilde{G}(q) = \tilde{\tau} + \tilde{\tau}_e, \tag{45}$$

where the definitions of $\tilde{M}(q)$, $\tilde{C}(q,\dot{q})$, $\tilde{G}(q)$, $\tilde{\tau}$ and $\tilde{\tau}_e$ can be referred to [9]. They are the transformations of Inertia matrix, Coriolis term, Gravitational term, control input and the external torque respectively. We will denote $\tilde{M}(q)$ as $\tilde{M}$, $\tilde{C}(q,\dot{q})$ as $\tilde{C}$ and $\tilde{G}(q)$ as $\tilde{G}$ for brevity.

**Theorem 5.** *The following control law almost globally asymptotically tracks $g_d(t)$ for a dynamical system described by* (45) *with the elastic force $f$ described by* (40)

$$\tilde{\tau} = \tilde{M}\dot{V}_d^* + \tilde{C}V_d^* + \tilde{G} + \tilde{M}(K_D e_V - f). \tag{46}$$

*Proof.* This control law achieves the following error dynamics

$$\dot{e}_V + K_D e_V - f(g_e) = 0 \tag{47}$$

We will consider the following positive definite Lyapunov candidate function

$$\begin{aligned} W &= \Psi + \tfrac{1}{2} e_V^T e_V \\ \implies \dot{W} &= \dot{\Psi} + e_V^T \dot{e}_V = -e_V^T K_D e_V \leq 0. \end{aligned} \tag{48}$$

The elastic forces in (43) and (46) are invariant to left translations i.e., when both the scene and the end effector's pose are left-translated by the same, the elastic force doesn't change as the elastic force is only a function of error configuration and from the way it was defined in (13) and (20), error configuration was shown to be left invariant in (23). It can also be seen that the the error dynamics in (44) and (47) only depend on the error configuration and the error velocities and have no dependence on the physical properties of the robot. Therefore, control laws designed for robots with the same kinematic structure can be transferred from one robot to the other without the hassle of tuning the potentials afresh. This becomes extremely handy when mimicking expert demonstrations as smaller robots can be trained with human demonstrations and can be seamlessly transferred to larger robots.

## VII. Training the Neural Network

Once the structure of the potential function is finalized (by fixing the number of layers and their sizes), an objective according to the needs of the user can be specified which can be posed as an optimization problem of minimizing a loss function by gradient descent. We consider an LQR-style problem where we have a running cost along the trajectory that needs to be minimized. A sample loss function for an error trajectory is shown in (49) with a positive weight $\lambda$. We could also add weighting matrices like in the LQR problem.

$$L = \int_0^T (\|\mathcal{I}_3 - R_e(t)\|_F^2 + \lambda \|\Omega_e(t)\|_2^2) dt \tag{49}$$

The procedure to shape the potential function to minimize this loss is shown in Fig. 3. We first start by forming a set of initial conditions around which the system is expected to start. Since any potential with random initialization (of course with some non-negative weights which can be taken care of by softplus function in PyTorch) becomes a valid potential function, we can obtain the corresponding stabilizing elastic force and integrate the system forward to obtain the error trajectories for each of the initial conditions for a user-defined fixed time $T$ without diverging trajectories. Here, we have used the error dynamics (44), but the other error dynamics introduced can be used. A mean loss is computed by taking the average of individual losses corresponding to the trajectory for each initial condition. Next, standard back-propagation algorithms (e.g., ADAM) can be used to update the parameters of the potential function. The function $h(\cdot)$ is used to represent the choice of our optimizer. This updated potential function generates the updated elastic force, and the system is integrated forward for all the initial conditions again. This cycle is repeated till convergence or any other user-specified termination criterion. Note that since the dynamics evolve on manifolds, a variational integrator like [26], [27] could be a better choice to integrate the system forward as they preserve the geometry of the manifold. The simplest integrator is an Euler integrator for the velocities and uses the velocities with the exponential map (or other methods consistent with the kinematics for non-Lie groups) to find the next point on the manifold. Euler integration for (44) for a step size $\Delta t$ is as follows

$$\begin{aligned} \Omega_e(t+\Delta t) &= \Omega_e(t) + (-K_D\Omega_e(t) + f_\theta(R_e))\Delta t \\ R_e(t+\Delta t) &= R_e(t) e^{\hat{\Omega}_e(t)\Delta t}. \end{aligned} \tag{50}$$

In Fig.3, the dynamical equations in the integration block

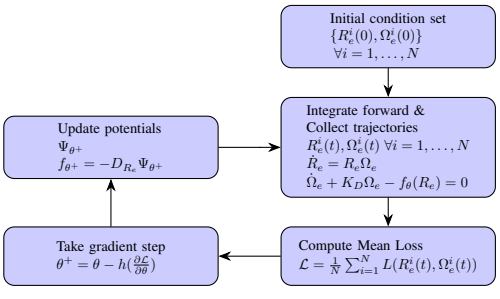



Fig. 3: Flow chart showing the training process.

can be replaced by (47) and updating the damping gain $K_D$, loss function appropriately to account for SE(3) trajectories. Though the positive definite damping matrix $K_D$ is shown to be fixed, it can also be treated as a decision variable in the optimization problem which would ease the restriction further and provide better performance.

## VIII. Simulation Results

### A. Implementation on SO(3) manifold

We consider the problem of regulating the rotation error $R_e$ to $\mathcal{I}_3$ for the dynamics in (44). If the error is asymptotically brought to $\mathcal{I}_3$ and the error velocity to 0, we are able to track any desired trajectory. A neural network with 3 hidden layers of 14 neurons each, using Kaiming initialization within bounds $(-\sqrt{3}, \sqrt{3})$, was trained on 50 initial conditions. Initial conditions were set with rotation errors normally distributed around $\pi$ radians from the identity matrix according to (51) and angular velocity errors with zero mean and unit variance. The training utilized the Adam optimizer with a learning rate of 0.01 over 3000 epochs to minimize the loss function in (49) with $\lambda = 1$ and $K_D = \mathcal{I}_3$. To visualize rotation errors and error velocities, plots of $\Psi_F(R_e(t)) = \frac{1}{2}tr[\mathcal{I}_3 - R_e(t)]$ and $\|e_V(t)\|_2^2$ are presented. Results in Fig.4 illustrate the evolution of rotation error trajectories for neural network, logarithmic, and Frobenius norm potential functions. Despite being designed for large initial errors around $\pi$ radians, performance for small errors is comparable across potential functions, as shown in Fig.4(c). We can clearly see the jumps due to discontinuities in the logarithmic potential function based controller and sluggish response of the Frobenius norm controller. The neural network controller achieved a fast response while maintaining smoothness.

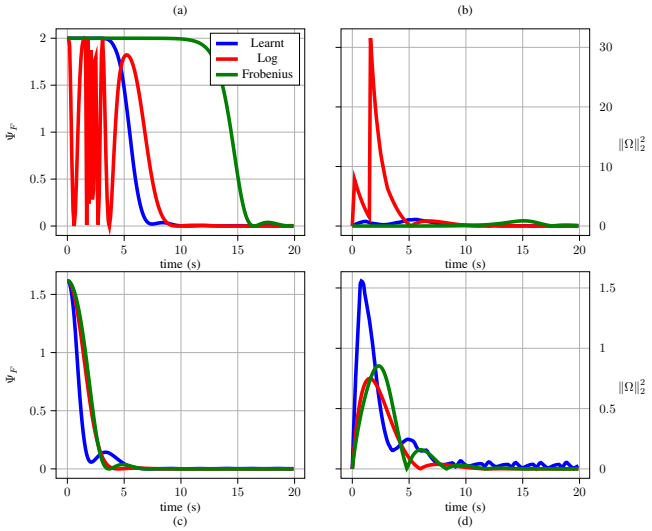



Fig. 4: Comparison of responses with various potential functions (a) $R_e$ trajectories for $0.99\pi$ radians initial error (b) and corresponding angular velocities errors. (c) $R_e$ trajectories for $0.1\pi$ radians initial error and (d) corresponding angular velocity errors.

### B. Implementation on SE(3) manifold

We will first consider the problem of regulating the configuration error on SE(3) i.e., $g_e$ to $\mathcal{I}_4$ for the error dynamics in (47). We change the first layer of the neural network to account for the increase in dimension from SO(3) to SE(3) but the remaining layers and other aspects of training remain same as VIII-A. The results are shown in Fig. 5.

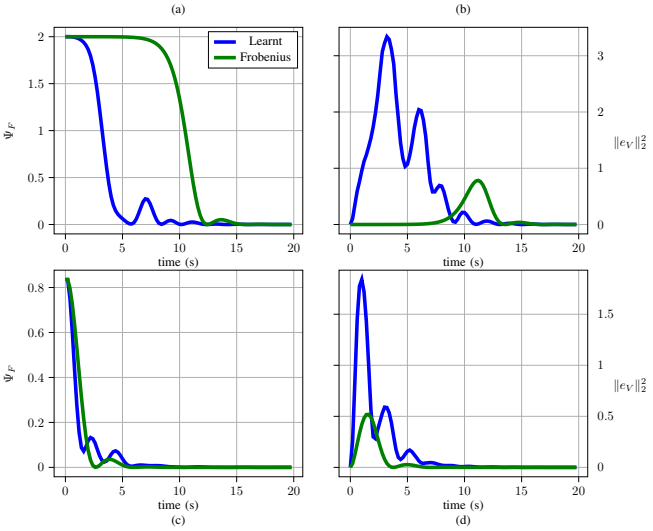



Fig. 5: Comparison of responses with various potential functions on SE(3) for error dynamics in (47) (a) $g_e$ trajectories for $0.99\pi$ radians initial rotation error and $p_e = [0.1529, -1.4135, 1.9227]$ (b) and corresponding angular velocities errors. (c) $g_e$ trajectories for $0.1\pi$ radians initial rotation error and $p_e = [-0.2083, 0.0364, 0.9568]$ (d) and corresponding angular velocity errors. $\Psi_F = \frac{1}{2}\|\mathcal{I}_4 - g_e\|_F^2$.

Next, the approach has been implemented on a UR5e robotic manipulator in the Mujoco environment (see Fig. 6). To show the advantages of the proposed approach, we will compare the regulation performance under large initial error conditions for the proposed approach and the benchmark approach, the geometric impedance control [9] with Frobenius norm-based potential function. The goal pose $g_d = (R_d, p_d)$ of the end-effector is given as

$$p_d = \begin{bmatrix}0.7 & 0.0 & 0.4\end{bmatrix}^T, \quad R_d = \begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & -1\end{bmatrix}$$

The initial condition $g_i = (R_i, p_i)$ is given as

$$p_i = p_d, \quad R_i = \text{Rot}(w, \theta)R_d, \tag{51}$$

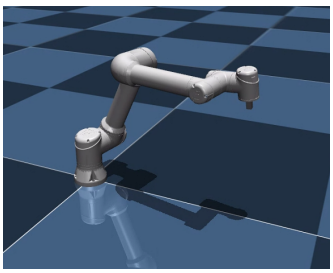


Fig. 6: UR5e robot manipulator implemented in Mujoco environment.

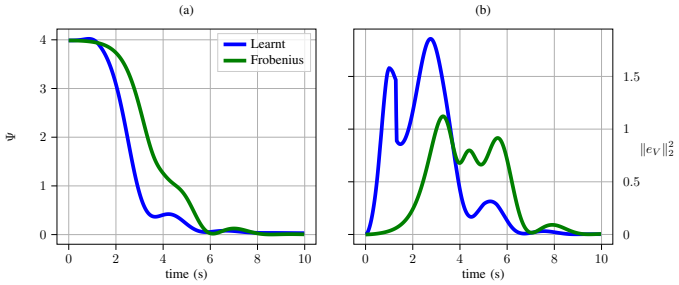



Fig. 7: (a) Potential function $\Psi$ with respect to time, (b) 2-norm of velocity errors, for the learning-based controller and the Frobenius-norm-based controllers are presented.

where $w = [-1, 1, 1]^T$, $\theta = \pi - \varepsilon$ with $0 < \varepsilon (= 0.001) \ll 1$, and $\text{Rot}(w, \theta)$ is an axis-angle representation of the rotation matrix, e.g., one can use Rodrigues formula to convert it into the rotation matrix. The results of this large initial rotational angle scenario are shown in Fig. 7. The proposed approach showed faster convergence compared to the Frobenius-norm-based controller.

Notice that we have considered the dynamics (45) of the robot's end effector which is assumed to move freely as an unconstrained rigid body in the free space despite having limitations on the configuration space due to physical limitations of the robot. The control law designed is similar in idea to a feed back linearization control law where we cancelled out some of the terms to make the error dynamics appear in a desired way and the elastic force only affected the final error dynamics without taking into consideration the limitations of the robot. Due to this, it is possible that the elastic force generated by the neural network potential might demand the robot's end effector to attain configurations that are unachievable (when the jacobian becomes singuar). This is an issue specific to the manipulator whereas the issue does not arise when we consider the control of an unconstrained rigid body on SE(3). It must be noted that this isn't a limitation of this particular approach alone, but is true for all potential function based control designs. To overcome this, one needs to incorporate constraints into the control law which is left for future work.

## IX. Conclusions

In this paper, a generic design methodology for designing trainable invariant potential functions using neural networks for dynamical systems evolving on manifolds has been presented. A detailed methodology to obtain the corresponding elastic forces through variations has been presented. Extensive analysis of the potential functions and stability for various manifolds of interest has also been presented. Finally, the methodology was demonstrated on two problems - a satellite orientation control and a robotic manipulator variable impedance control and the results were presented along with comparisons with benchmark controllers.

## References


[1] T. Lee *et al.*, "Geometric tracking control of a quadrotor uav on SE(3)," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5420–5425.

[2] T. Lee, "Exponential stability of an attitude tracking control system on so (3) for large-angle rotational maneuvers," *Systems & Control Letters*, vol. 61, no. 1, pp. 231–237, 2012.

[3] T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear robust tracking control of a quadrotor uav on se (3)," *Asian journal of control*, vol. 15, no. 2, pp. 391–408, 2013.

[4] P. Kotaru, R. Edmonson, and K. Sreenath, "Geometric l1 adaptive attitude control for a quadrotor unmanned aerial vehicle," 2020.

[5] P. Kotaru, G. Wu, and K. Sreenath, "Dynamics and control of a quadrotor with a payload suspended through an elastic cable," in *2017 American Control Conference (ACC)*, 2017, pp. 3906–3913.

[6] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.

[7] N. Hogan, "Impedance control: An approach to manipulation: Part ii—implementation," 1985.

[8] J. Seo, N. P. Prakash, X. Zhang, C. Wang, J. Choi, M. Tomizuka, and R. Horowitz, "Contact-rich se (3)-equivariant robot manipulation task learning via geometric impedance control," *IEEE Robotics and Automation Letters*, 2023.

[9] J. Seo *et al.*, "Geometric impedance control on SE(3) for robotic manipulators," *IFAC World Congress 2023, Yokohama, Japan*, 2023.

[10] J. Seo, N. P. S. Prakash, J. Choi, and R. Horowitz, "A comparison between lie group- and lie algebra- based potential functions for geometric impedance control," 2024.

[11] F. Bullo and R. M. Murray, "Proportional derivative (PD) control on the euclidean group," 1995.

[12] F. Bullo and R. Murray, "Tracking for fully actuated mechanical systems: a geometric framework," in *Automatica*, vol. 35, no. 1. Elsevier, 1999, pp. 17–34.

[13] F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," 1995.

[14] F. C. Park, J. E. Bobrow, and S. R. Ploen, "A lie group formulation of robot dynamics," *The International journal of robotics research*, vol. 14, no. 6, pp. 609–618, 1995.

[15] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," 2000.

[16] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *International Conference on Machine Learning*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:15898758

[17] G. Manek and J. Z. Kolter, "Learning stable deep dynamics models," *CoRR*, vol. abs/2001.06116, 2020. [Online]. Available: https://arxiv.org/abs/2001.06116

[18] N. Potu Surya Prakash, J. Seo, K. Sreenath, J. Choi, and R. Horowitz, "Variable impedance control using deep geometric potential fields," in *Modeling, Estimation and Controls Conference*, October 2024.

[19] N. Potu Surya Prakash, "Learning and optimization methods for robust control of hard disk drives and geometric control of fully actuated mechanical systems," Ph.D. dissertation, University of California, Berkeley, 2024.

[20] T. Lee, M. Leok, and H. N. McClamroch, "Global formulations of lagrangian and hamiltonian dynamics on manifolds," vol. 13, 2017.

[21] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[22] P. Kotaru, G. Wu, and K. Sreenath, "Differential-flatness and control of quadrotor(s) with a payload suspended through flexible cable(s)," in *2018 Indian Control Conference (ICC)*, 2018, pp. 352–357.

[23] J. Zeng, P. Kotaru, and K. Sreenath, "Geometric control and differential flatness of a quadrotor uav with load suspended from a pulley," in *2019 American Control Conference (ACC)*, 2019, pp. 2420–2427.

[24] J. Behrmann, W. Grathwohl, R. T. Q. Chen, D. Duvenaud, and J.-H. Jacobsen, "Invertible residual networks," 2019.

[25] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[26] M. Kobilarov, K. Crane, and M. Desbrun, "Lie group integrators for animation and control of vehicles," *ACM Trans. Graph.*, vol. 28, no. 2, may 2009.

[27] N. P. S. Prakash, "A computational approach for variational integration of attitude dynamics on so(3)," 2022.