University of Sheffield | School of Mechanical, Aerospace & Civil Engineering

Geometry-Informed Systems for Robotic Manipulation

Methods in Planning & Control

Eytan Canzini[1,2]

[1]Digital Manufacturing Group (DMG)
School of Mechanical, Aerospace & Civil Engineering
University of Sheffield

[2]Airbus Robotics R&T UK

July 16, 2025

Outline

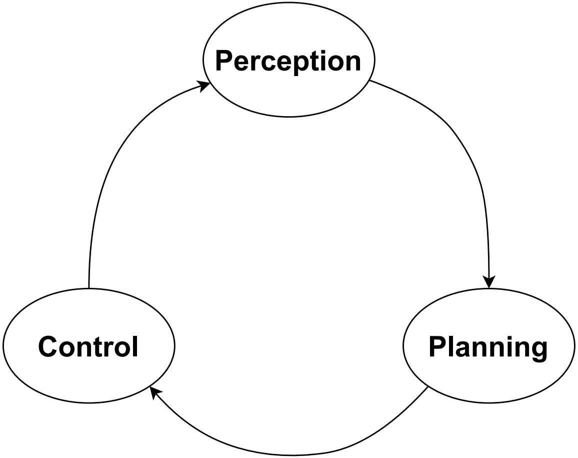
1. A Brief Introduction to Robotic Manipulation

2. Riemannian Manifolds & The $\mathbb{SE}(3)$ Matrix Lie Group

3. Continuous Constrained Planning

4. Geometric Control of Robotic Systems

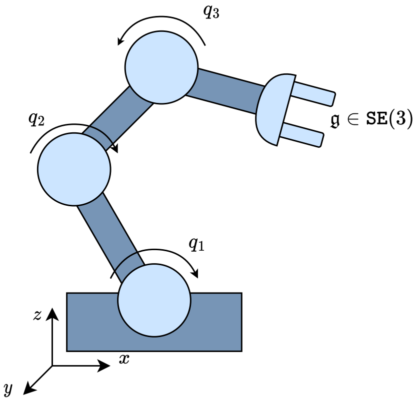
The Feedback Loop of Robotics



- Perception provides estimates of the current state $\mathbf{x}_t$ of the robot
  - Covers the way a robot "visualises" its environment
  - Coupled with filtering for state estimation
- Planning chooses how to go from the current state $\mathbf{x}_t$ to a target state $\mathbf{x}_T$
  - Can either plan for an entire trajectory $\mathbf{x}_{1:T}$...
  - ...or just the next state $\mathbf{x}_{t+1}$
- Control moves the robot from current state $\mathbf{x}_t$ to the next state $\mathbf{x}_{t+1}$

University of Sheffield | School of Mechanical, Aerospace & Civil Engineering

Robot Manipulators



- Robot manipulators are a set of n joints organised either sequentially or in parallel
- Set of rotational or prismatic elements $\mathfrak{q} = [q_1, \ldots, q_n] \in \mathcal{Q}$, where transformation from the base frame to the tool frame is given by the forward kinematics

$$\mathfrak{g} = \boldsymbol{g}(\mathfrak{q}) = \mathbf{T}_n^0(\mathfrak{q}) = \mathbf{T}_1^0(q_1)\mathbf{T}_2^1(q_2)\cdots\mathbf{T}_n^{n-1}(q_n)$$

and velocities by the Jacobian

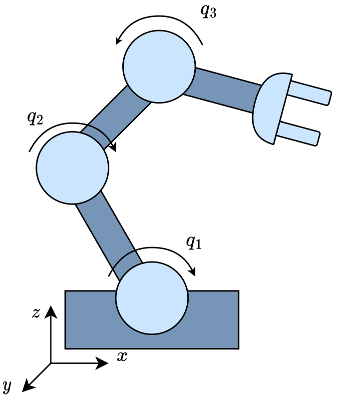
$$\boldsymbol{\xi} = \mathbf{J}(\mathfrak{q})\dot{\mathfrak{q}}$$

- In the joint frame, the manipulator dynamics are

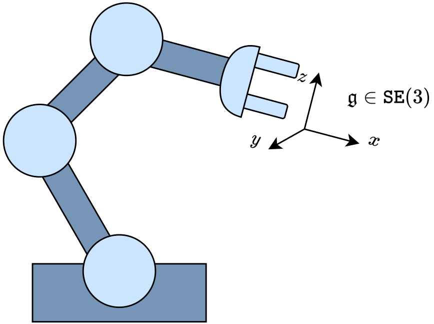
$$\mathbf{M}(\mathfrak{q})\ddot{\mathfrak{q}} + \mathbf{C}(\mathfrak{q}, \dot{\mathfrak{q}})\dot{\mathfrak{q}} + \mathcal{G}(\mathfrak{q}) = \tau$$

Joint Space vs. Operational Space

Joint Space Formulation



Operational (Task) Space Formulation



$$\mathfrak{q} \underset{\boldsymbol{g}^{-1}(\mathfrak{q})}{\overset{\boldsymbol{g}(\mathfrak{q})}{\rightleftarrows}} \mathfrak{g}$$

$$\dot{\mathfrak{q}} \underset{\mathbf{J}^{+}(\mathfrak{q})}{\overset{\mathbf{J}(\mathfrak{q})}{\rightleftarrows}} \boldsymbol{\xi}$$

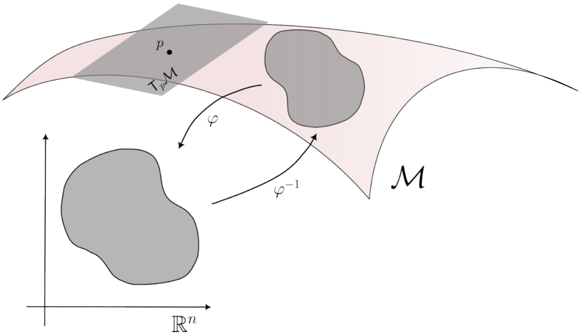
$$\mathbf{x} = (\mathfrak{q}, \dot{\mathfrak{q}}) \in \mathsf{T}\mathcal{Q}$$

$$\mathbf{x} = (\mathfrak{g}, \boldsymbol{\xi}) \in \mathsf{T}\mathtt{SE}(3)$$

$$\mathbf{M}(\mathfrak{q})\ddot{\mathfrak{q}} + \mathbf{C}(\mathfrak{q}, \dot{\mathfrak{q}})\dot{\mathfrak{q}} + \mathcal{G}(\mathfrak{q}) = \tau$$

$$\boldsymbol{\Lambda}(\mathfrak{g})\ddot{\mathfrak{g}} + \mu(\mathfrak{g}, \dot{\mathfrak{g}})\dot{\mathfrak{g}} + \mathcal{P}(\mathfrak{g}) = F$$

Riemannian Manifolds



- A Riemannian Manifold is a geometric space $\mathcal{M}$ imbued with a Riemannian metric M creating the pair $(\mathcal{M}, M)$
- The distance between two points $p_0, p_1 \in \mathcal{M}$ is found through the geodesic

$$d(p_0, p_1) = \inf_{\gamma \in PC([0,1];\mathcal{M})} \int_I \sqrt{\dot{\gamma}^\top M \dot{\gamma}}\, \mathrm{d}t$$

for all piece-wise curves $\gamma : I \to [0, 1]$.

University of Sheffield | School of Mechanical, Aerospace & Civil Engineering

The $\mathtt{SE}(3)$ Matrix Lie Group

The $\mathtt{SE}(3)$ matrix Lie group is defined by positions and orientations:

$$\mathfrak{g} \triangleq \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \mathtt{SE}(3) \,\middle|\, \mathbf{R} \in \mathtt{SO}(3),\ \mathbf{p} \in \mathbb{R}^3 \right\}$$

When following a trajectory across the group $t \mapsto \boldsymbol{\xi} = (\boldsymbol{v}, \boldsymbol{\omega}) \in \mathbb{R}^3 \bigoplus \mathbb{R}^3$, the isomorphism $\hat{(\cdot)} : \mathbb{R}^3 \bigoplus \mathbb{R}^3 \to \mathfrak{se}(3)$ provides us with the definition of twists:

$$\hat{\boldsymbol{\xi}} \triangleq \left\{ \begin{bmatrix} \hat{\boldsymbol{\omega}} & \boldsymbol{v} \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix} \in \mathfrak{se}(3) \,\middle|\, \hat{\boldsymbol{\omega}} \in \mathfrak{so}(3),\ \boldsymbol{v} \in \mathbb{R}^3 \right\}$$

The kinematics of $\mathtt{SE}(3)$ provide us with a left-invariant vector velocity:

$$\dot{\mathfrak{g}} = \mathsf{T}_{\mathsf{e}}\mathsf{L}_{\mathfrak{g}}(\hat{\boldsymbol{\xi}}) = \mathfrak{g}\hat{\boldsymbol{\xi}}$$

Why use Geometric Kinematics & Dynamics?

- Most tasks for manipulators define some error in the task space $\mathtt{SE}(3)$ so we can decompose the rotational and position errors on the $\mathtt{SE}(3)$ group
- Generalisation of tasks - rather than making the solution task dependent, focus on defining the geometry
- Leverage elements of geometry to compute paths through the use of isomorphisms in lower dimensional spaces
- Many dynamics formulations rely on geometric roots - use the tools that are present rather than make new ones!

Geometric Kinematics of Manipulators

The forward kinematics of the robot manipulator $\boldsymbol{g}(\mathfrak{q})$ can be represented using the product of exponentials formula

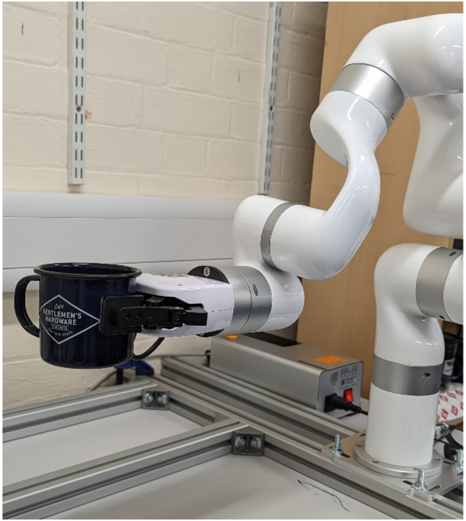
$$\boldsymbol{g}(\mathfrak{q}) = e^{\hat{\boldsymbol{\xi}}_1 q_1} e^{\hat{\boldsymbol{\xi}}_2 q_2} \dots e^{\hat{\boldsymbol{\xi}}_n q_n} \boldsymbol{g}(\mathbf{0})$$

which produces a set of screw rotations around each joint.
The geometric body Jacobian for a manipulator in task space is defined as

$$\begin{aligned} \boldsymbol{\xi}_b =& \mathbf{J}_b(\mathfrak{q})\dot{\mathfrak{q}} \\ \text{where } \mathbf{J}_b(\mathfrak{q}) =& \begin{bmatrix} \boldsymbol{\xi}_1^{\dagger} & \cdots & \boldsymbol{\xi}_{n-1}^{\dagger} & \boldsymbol{\xi}_n^{\dagger} \end{bmatrix} \\ \boldsymbol{\xi}_i^{\dagger} =& \mathrm{Ad}_{\boldsymbol{g}(\mathfrak{q})}^{-1} \boldsymbol{\xi}_i \end{aligned}$$

Each column of $\mathbf{J}_b$ corresponds to a joint twist in the tool frame

Constrained Manipulation



- Constrained robot trajectories limit the manipulability of end-effectors in $\mathtt{SE}(3)$
- Permitted movements in $\mathtt{SE}(3)$ are represented as a vector

$$\mathbb{C} = \begin{bmatrix} c_x & c_y & c_z & c_\beta & c_\alpha & c_\gamma \end{bmatrix}$$

where a binary value $c \in [0, 1]$ indicates whether a dimension is constrained

- A constraint function $\mathbf{f}(\mathfrak{q})$ indicates whether a pose is valid

$$\begin{aligned} \mathbf{f}(\mathfrak{q}) &= \mathbb{C} \odot \mathfrak{g} \\ &= \mathbb{C} \odot \mathbf{T}_n^0(\mathfrak{q}) \end{aligned}$$

when $\|\mathbf{f}(\mathfrak{q})\| = 0$.

Why is constrained planning hard?

- The goal of path planning is to find the shortest optimal trajectory τ^* between a start and end pose

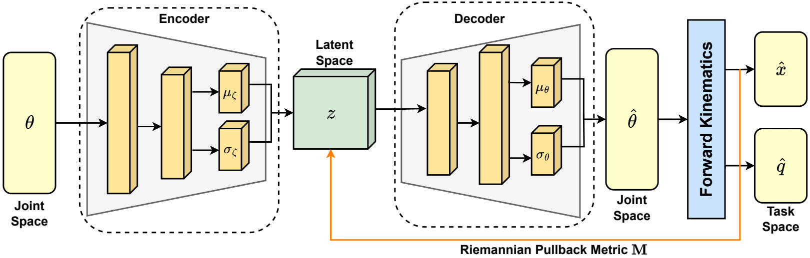
$$\tau* = \arg\min_{\tau} \sum_{i=1}^{n} J(\tau_i)$$

- Current approaches utilise sampling-based planners which generate graphs of points that satisfy the constraints
- Constraints are sampled at every time-step in the planning process
- More complex constraints = More time generating paths
- Constrain in task space, plan in joint space - can we link the constraints to the joints?
- Planning in higher dimensions - can we use a lower dimensional representation

Learning Manifolds from Data

Using a data-driven approach, we can model $\mathcal{M}$ as a stochastic function $f : \mathcal{Z} \rightarrow \mathcal{X}$

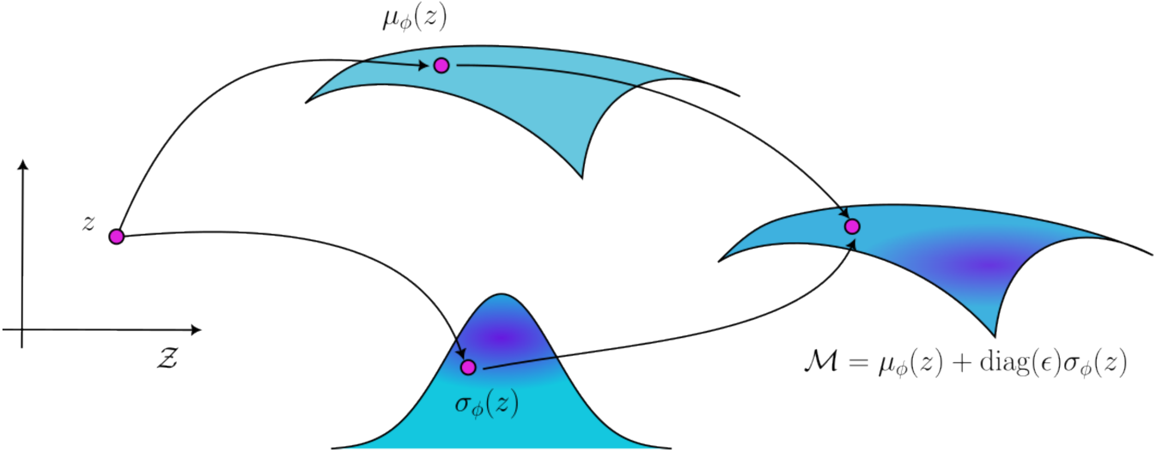
$$\mathcal{M} = f_{\phi}(z) = \mu_{\phi}(z) + \text{diag}(\epsilon)\sigma_{\phi}(z), \forall z \in \mathcal{Z} \text{ and } \dim\mathcal{Z} < \dim\mathcal{X}$$



The Riemannian metric is found from the reconstruction loss of the VAE when going from the latent space z to $\mathrm{SE}(3)$:

$$M^{\mathfrak{q}}(z) = \mathbf{J}^{\mathfrak{q}}_{\mu}(z)^{\intercal}\mathbf{J}^{\mathfrak{q}}_{\mu}(z) + \mathbf{J}^{\mathfrak{q}}_{\sigma}(z)^{\intercal}\mathbf{J}^{\mathfrak{q}}_{\sigma}(z)$$

Latent Space Manifolds



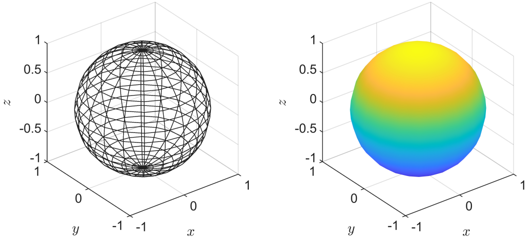
Beik-Mohammadi et al., "Reactive motion generation on learned Riemannian manifolds", IJRR, 2023, doi: 10.1177/02783649231193046

Constraint Manifolds as Sub-Manifolds

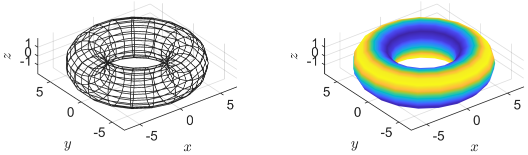
- Constraint manifolds don't alter the geometry of the manifold
- Embedded sub-manifolds can be represented as constraints by an inclusion metric $\iota_{\mathfrak{q}}$
- Using the constraint function $\mathbf{f}(\mathfrak{q})$, we construct a set of constraint sub-manifolds

$$\tilde{\mathcal{M}}_\theta = \{\tilde{\mathcal{M}}_i \subseteq \mathcal{M} \mid \iota_\theta\},$$

where $\tilde{\mathcal{M}}_i := \{z \in \mathcal{Z} \mid f(z)\}$, for $i = 1, \ldots, N$.



(a) $\mathcal{S}^2$ embedded in $\mathbb{R}^3$



(b) $\mathbb{T}^2$ embedded in $\mathbb{R}^3$

Continuous Paths on Learned Manifolds

In our latent space, the geodesic distance can be approximated by a spline

$$\gamma(z) \approx \left[\sum_{n=1}^{2} \left(\sum_{k=1}^{K} \varphi_{n,k} t^k \right) + \varphi_{n,0} \right]$$

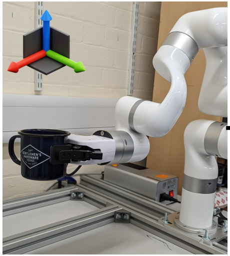
Finding the values of φ can be framed as an optimisation problem parameterised by a neural network with parameters θ

$$\theta^* = \arg\max_{\theta \in \Theta} \min_{\gamma(z)} \mathcal{L}_\gamma$$

where $\mathcal{L}_\gamma = \int_I \sqrt{\dot{z}_1^\intercal M \dot{z}_1 + \dot{z}_2^\intercal M \dot{z}_2}\, \mathrm{d}t$. This can be solved using reinforcement learning as policy search.

University of Sheffield | School of Mechanical, Aerospace & Civil Engineering

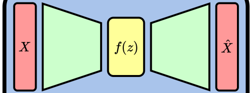
Constrained Planning on Learned Manifolds



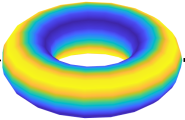
Robot kinematic model $\mathbf{T}_e^0$

Task constraint Information $\mathbb{C}$

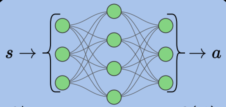
Robot Motion Constraint Manifold Learner



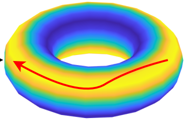
$\tilde{\mathcal{M}} \subseteq \mathcal{M}$



Policy Search Geodesic Solver

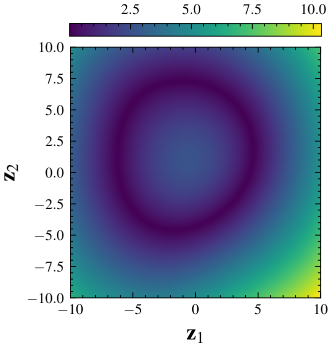


$$\theta^* = \underset{\theta \in \Theta}{\text{argmax}} \ \underset{\phi}{\min} \ \mathcal{L}(\phi)$$

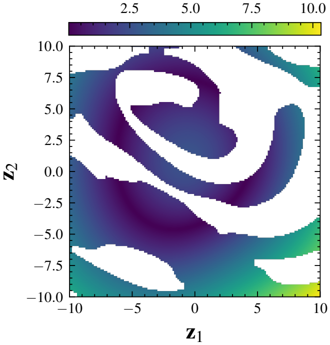


Continuous optimal trajectory τ^* that adheres to on-manifold constraints

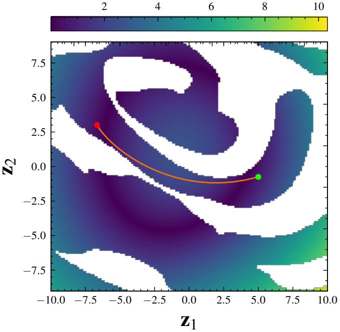
Resulting Manifold & Path



$\langle \mathcal{M}, \mathbf{M} \rangle$



$\tilde{\mathcal{M}} \subseteq \mathcal{M}$



$\gamma(z)$

Constrained Planning in Action

Click to Play Video

Future Directions

- Learning manifolds requires a lot of data
- Disconnection between sub-manifolds - what happens when regions are not linked?
- Non-linearities lead to manipulator singularities - inverse kinematics don't have unique solutions
- Replanning under uncertainty - improving real-time performance
- Embedding obstacle avoidance - modifying the Riemannian metric to accommodate non-stationary environments

Visual Servoing Control

- Visual servoing finds control inputs based on comparing current visual features against target features

$$e(t) = s(m(t), a) - s^*$$

- The target features are also associated with a target pose $\mathfrak{g}^* \in \mathrm{SE}(3)$
- Hand-crafting features is a time-consuming process that relies on user knowledge

Click to Play Video

Sciliano and Khatib, "Handbook of Robotics", 2007, Springer

Depth Clouds

A depth cloud P is a finite set of points, where each point relates to a camera pixel in the resolution range $[\mathbb{X}^n, \mathbb{Y}^m]$ and the distance estimate:

$$P = \left\{ [x, y, Z] \in \mathbb{R}^3_+ \,|\, x \in \mathbb{X}^n, y \in \mathbb{Y}^m, Z \sim f(x, y) \right\}$$

Optimal Transport of Depth Clouds

Both the initial and target depth clouds ρ_0, ρ_1 exist in a Wasserstein space:

$$\mathcal{P}_p(\mathtt{SE}(3)) = \left\{ \rho \in \mathcal{P}(\mathtt{SE}(3)) : \int_{\mathtt{G}} d^p(\mathfrak{g}_0, \mathfrak{g}_1)\, \mathrm{d}\rho(\mathfrak{g}) < +\infty \right\}, \quad \forall p \geq 1$$

This space is imbued with a p-Wasserstein distance for unbalanced measures:

$$\mathbb{W}_\tau^p(\rho_0, \rho_1) = \left(\inf_{\pi \in \Pi(\mathtt{G}\times\mathtt{G};\mu,\nu)} \int_{\mathtt{G}\times\mathtt{G}} d^p(x,y)\mathrm{d}\pi(x,y) + \tau\mathbb{D}_\varphi(P_{1,\sharp}\pi|\rho_0) + \tau\mathbb{D}_\varphi(P_{2,\sharp}\pi|\rho_1) \right)^{\frac{1}{p}}$$

In $\mathtt{SE}(3)$, the Riemannian metric is the Hamiltonian metric tensor $\mathbb{G}_{\mathcal{H}} = \begin{bmatrix} \mathbb{G}_{\mathbf{p}} & \mathbf{0} \\ \mathbf{0} & \mathbb{G}_{\mathbf{R}} \end{bmatrix} = (\mathbf{J}_b^{+\top}(\mathfrak{q})\mathbf{M}(\mathfrak{q})\mathbf{J}_b^{+}(\mathfrak{q}))^{-1}$ with an approximate geodesic:

$$d(\mathfrak{g}_0, \mathfrak{g}_1) \approx \sqrt{\delta_{\mathbf{R}}(\mathbf{R}_0, \mathbf{R}_1) + \delta_{\mathbf{p}}(\mathbf{p}_0, \mathbf{p}_1)}$$

where $\delta_{\mathbf{R}}(\mathbf{R}_0, \mathbf{R}_1) = \|\mathbb{G}_{\mathbf{R}}\|_F \cos^{-1}\left(\frac{\mathrm{tr}((\mathbf{R}_0^\top \mathbf{R}_1)) - 1}{2}\right)$, $\delta_{\mathbf{p}}(\mathbf{p_0}, \mathbf{p}_1) = \|\mathbb{G}_{\mathbf{P}}(\mathbf{p}_0 - \mathbf{p}_1)\|_2$

Dynamic Optimal Transport

In the "Eulerian" sense, the flow of the depth cloud on $\mathtt{SE}(3)$ is governed by the Hamilton-Jacobi continuity equation

$$\frac{\partial \rho_t}{\partial t} + \nabla(\rho_t \boldsymbol{\xi}) = s_t$$

This produces a set of curves in the Wasserstein space

$$\bar{\mathcal{C}}(\rho_0, \rho_1) = \left\{ (\rho_t, (\rho_t \boldsymbol{\xi}), s_t) : \frac{\partial \rho_t}{\partial t} + \nabla(\rho_t \boldsymbol{\xi}) = s_t, \rho_{t=0} = \mu, \rho_{t=1} = \nu \right\}$$

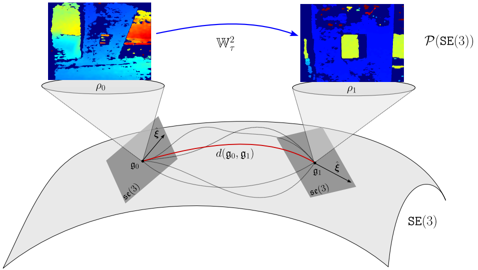
Then the p-Wasserstein distance becomes

$$\mathbb{W}_\tau^2 = \min_{(\rho_t, (\rho_t \boldsymbol{\xi}), s_t) \in \bar{\mathcal{C}}(\rho_0, \rho_1)} \int_0^1 \int_{\mathtt{G}} d^p(\mathfrak{g}, \mathfrak{g}_1) + \tau \Theta(\rho_t, s_t) \mathrm{d}\mathfrak{g} \, \mathrm{d}t$$

This becomes a stochastic optimal control problem!

Y. Chen et al., "Optimal Transport Over a Linear Dynamical System," IEEE T-AC, May 2017, doi: 10.1109/TAC.2016.2602103

Depth Measures on Lie Groups



For a manipulator with geometric dynamics in the form:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

The input $\mathbf{u} = \mathbf{B}^+ \begin{bmatrix} \boldsymbol{v}_t \\ \boldsymbol{\omega}_t \end{bmatrix}$ constitutes a wrench in $\mathrm{SE}(3)$.

Given a controllability Gramian:

$$W_c = \int_0^1 \Psi(\lambda) B B^\top \Psi^{-1}(\lambda) \mathrm{d}\lambda$$

and a transport map T^* obtained from the solving the OT problem, the control wrench is denoted as:

$$u = B^\top \Psi^\top W_c^{-1} \left[T^\star \circ T_t^{-1} - T_t^{-1} \right]$$

with:

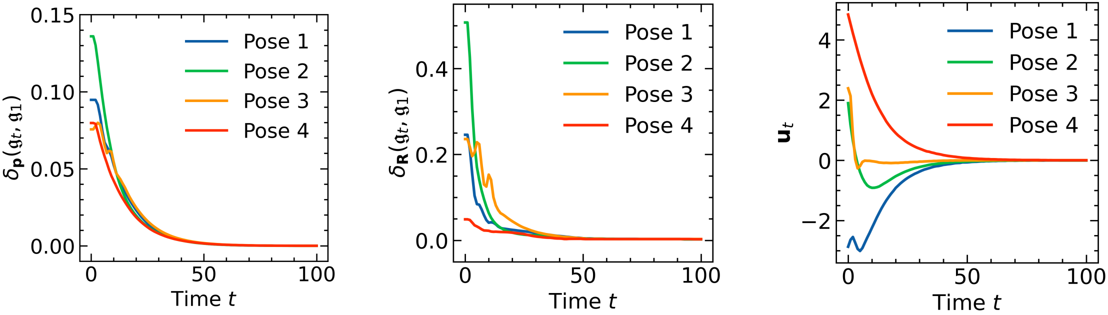
$$T_t = \Psi W_c(t,1) W_c^{-1} \Psi \mathbf{x} + W_c(0,t) \Psi^\top W_c^{-1} T^\star$$

Y. Chen et al., "Optimal Transport Over a Linear Dynamical System," IEEE T-AC, May 2017, doi: 10.1109/TAC.2016.2602103

Geometric Visual Servo Control...

Click to Play Video

Geometric Errors



University of Sheffield | School of Mechanical, Aerospace & Civil Engineering

...with some caveats

Click to Play Video

Summary

- Methods in differential geometry provide a rich framework to describe robotic manipulation tasks
- Applications beyond just kinematics - geometric dynamics can utilise geodesics for generating control inputs
- Singularities remain a threat - all stability criteria assume full rank Jacobian
- Dimensionality remains the limiting factor - point clouds can have 10^5 points!

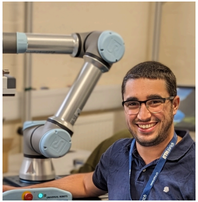
Thanks for listening!

Engineering and Physical Sciences Research Council

University of Sheffield | School of Mechanical, Aerospace & Civil Engineering

University of Sheffield | Centre for Machine Intelligence

AIRBUS



Eytan Canzini
School of Mech., Aero. & Civil Engineering

Dr. Marc Auledas-Noguera
School of Mech., Aero. & Civil Engineering

Dr. Simon Pope
School of Electrical & Electronic Engineering

Prof. Ashutosh Tiwari, *FREng*
School of Mech., Aero. & Civil Engineering

SCAN ME