Learnt skill manifolds for robot control

Ioannis Havoutis Subramanian Ramamoorthy

I. Motivation

Many humanoid robotic tasks of common interest require the agent to satisfy a number of complex skill specifications. For instance, the task of handling a tray in a typical service scenario requires the robot to achieve one of a certain set of goal states while respecting state constraints that might induce slipping, falling or collisions while also taking care of the manipulator dynamics and internal constraints. Everyday tasks, like tray handling, need to be defined over a nontrivially large domain but in the absence of models it is hard to define robust task-level controllers.

We develop a framework that uses manifolds to encode robotic skills. Such skill manifolds are learnt from demonstrated data and result in control strategies suitable for problems defined over large domains with strict skill specifications. In such scenarios planning and state evolution is restricted to geodesic trajectories, as states off the manifold fail the skill specifications implied by the demonstration data (e.g. Fig. 1).

II. Approach

Our framework of learning control strategies involves two components. Firstly, we encode the task in a *skill manifold*, a subspace in the underlying state space that is defined by the equivalence class of trajectories corresponding to various instances of the particular task. Then, we define cost hypersurfaces that penalize deviations from this subspace of states within the ambient space. This yields a vector field in the *ambient* space that constitutes both a basic plan from an initial condition and a controller to counteract unforeseen perturbations.

This gives a clearer interpretation to what the controller is achieving: enforcing a large domain vector field *towards* the manifold and *along* the manifold. This makes the consideration of obstacles [1] and disturbances much more natural, without having to worry about how they themselves may be mapped to an artificial low dimensional space.

Other methods have difficulty achieving large domain coverage as learning-based probabilistic approaches provide good local approximations but lack more global consideration of the system or task [2], [3], [4]. We can naturally handle this by considering the full space of the system, consisting of the union of the desirable subspace (manifold) where geodesic trajectories (need to) evolve, and the ambient space that surrounds the learnt manifold model.

I.Havoutis is with the *Advanced Robotics* lab within the *Italian Institute of Technology* {Ioannis.Havoutis@iit.it}, S.Ramamoorthy leads the *Robust Autonomy & Decisions Group* within the *Institute of Perception Action and Behaviour*, *School of Informatics*, *University of Edinburgh* {S.Ramamoorthy@ed.ac.uk}.

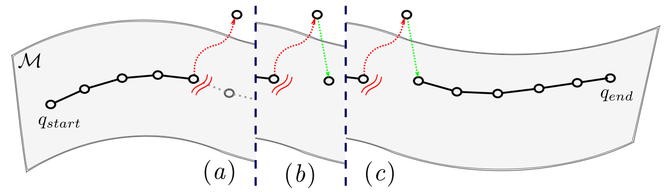


Fig. 1. A sketch of an example where the ability to project back to the desired space is necessary. (*a*) The system executes a geodesic trajectory when an unforeseen perturbation drives the state of the system to an off-manifold point. The remaining trajectory points are discarded. (*b*) Replanning from an off-manifold point would be insensible to the desired state evolution. Instead, we find the projection of the off-manifold state on the underlying geometry. This is the closest point that we then control for in a reactive manner. (*c*) A new geodesic trajectory is replanned, starting from the projection state and reaching to the goal state.

Our method provides a consistent metric that is used to evaluate the "*value*" of states in ambient space in a general *model-free* fashion (2(b)). We demonstrate efficient path planning, fast re-planning and online control in the face of relatively large scale perturbations for a simulated 3-link arm.

III. Manifold control on the 3-link arm

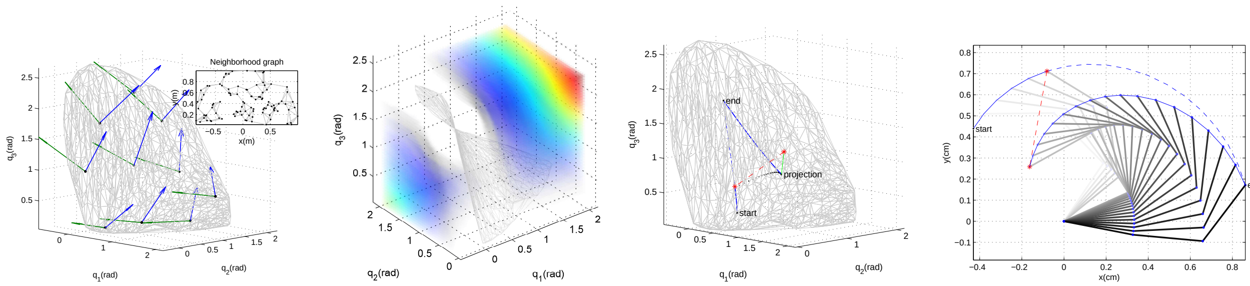
This example elucidates the basic concepts underlying our approach. With the *3-link planar* arm we can explicitly visualize both the configuration space and the manifold that, in this case, corresponds to a specific redundancy resolution strategy. The arm is a series of three rigid links, of $1/3$ length, that are coupled with hinge joints, producing a redundant system with 3 degrees of freedom that is constrained to move on a 2 dimensional plane.

We randomly sample 100 Cartesian points from the upper semicircle of the task space of the system. We run the task space dataset through an iterative redundancy resolution procedure that minimizes the weighted distance to a convenience (e.g., minimum strain) pose,

$$\min \|w(\mathbf{q}-\mathbf{q_c})\|^2 \, , \; \textit{subject to } f(\mathbf{q})-\mathbf{x}=0, \qquad (1)$$

where w is a weighting vector, f is the forward kinematics and $\mathbf{x}$ is the goal endpoint position on the plane, and get the corresponding joint space datapoints, $\mathbf{q}=(q_1,q_2,q_3)$.

The resulting $\mathbf{q}$'s trace a smooth nonlinear manifold in joint space, depicted in Fig. 2(a). This is the surface that the family of solutions belonging to the specific redundancy resolution strategy trace. Also different redundancy resolution strategies would produce different optimality manifolds. We note that, in general, this kind of information is not explicitly known (in the case of human demonstration) or visualizable, for many complex problems.



(a) Learnt manifold model. (b) Manifold metric. (c) Perturbed geodesic path. (d) Example in task space.

Fig. 2. (*a*) The manifold model learnt from data; the neighbourhood graph in task space (inset plot), and the learnt tangent space that the model predicts in the high dimensional space. The thin gray mesh surface is produced by densely sampling the manifold and helps convey a clearer perspective of the geometry of the space in question. (*b*) Volumetric plot of metric derived from our model. The metric evaluates the distance to the *modelled* surface and it smoothly surrounds the underlying manifold. Distances range from dark blue (small) to red (large), while the closest distances are completely transparent for clarity. (*c*),(*d*) A typical trajectory resulting from our approach. A geodesic path from start to end is computed, with a random perturbation occurring at time $t = 0.25$ that pushes the state away from the manifold. This new state is projected back on to the manifold to find the closest feasible state. A path from the projected point to the goal is then executed before continuing along. The dashed blue line is the initial predicted trajectory while the red line is the motion due to the (severe) perturbation occurring at the first red star. The state is then pushed away from the initial trajectory and a new path to the goal is replanned after the novel state is projected on the learnt manifold.

A. Implementation

We can see that the manifold can be naturally represented with a two dimensional tangent space, and we learn a model of the manifold, $\mathcal{H}_\theta$, as an approximation with Radial Basis Functions (RBFs). We can subsequently evaluate $\mathcal{H}_\theta$ at any point in our joint space. For example Figure 2(a) shows the learnt tangent basis approximation evaluated at the centres of the RBFs that build up the model. Note that the basis vectors are aligned and vary smoothly, i.e. we obtain good generalization within the region of support of the data. This way, in order to traverse the manifold we need to evaluate the learned tangent basis and follow each *local frame* for each consecutive step, in other words follow the blue and green arrows of Figure 2(a) for each point in question.

B. Results

To evaluate the accuracy of the model we randomly pick 100 start and end points and plan a trajectory between them, first with our method and second, with a naive quintic polynomial method as in Craig [5]. We distinguish two cases; an unperturbed trajectory, and a *random perturbation* occurring at $t = 0.25$ (Figure 2(c)). We calculate the average cost per trajectory and average over the results for each case (Table I). The results show that the use of the manifold achieves consistently lower cost trajectories, while the difference is multiplied in the case where a perturbation occurs. The interpretation being that the naive planner is forced to stay in a high cost patch of the state space while the manifold finds the appropriate short path to the cost-optimal surface.

IV. Conclusion

Integrated planning and control schemes need to accommodate sophisticated specifications arising at all levels from joint-level limits to global stability and other multivariate constraints, e.g., [6], [7]. We propose an approach that solves this problem by utilizing a learnt manifold and a correspondingly derived cost hypersurface, in a model free setting. The distance to the learnt manifold can be viewed as a metric of closeness to the desired *family* of solutions, while being able to directly compute the best feasible state given an arbitrary ambient state space point allows us to reactively accommodate unforeseen perturbations that drive the state of the system to the undesired (off-manifold) ambient space.

TABLE I

Mean costs evaluated against the true cost functions. Results are mean±standard deviation over sets of 100 random sampled trials.

System	*Method*	*Unperturbed traj. cost*	*Perturbed traj. cost*
3-link	*naive*	0.9239 ± 0.1799	1.401 ± 0.3610
	manifold	0.8724 ± 0.1723	1.214 ± 0.2597

References

[1] I. Havoutis and S. Ramamoorthy, "Constrained geodesic trajectory generation on learnt skill manifolds," *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.*, October 2010.

[2] E. Gribovskaya, K. Zadeh, S. Mohammad, and A. Billard, "Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators [accepted]," *International Journal of Robotics Research*, 2010.

[3] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 1398–1403.

[4] M. Pardowitz, S. Knoop, R. Dillmann, and R. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 322 –332, apr. 2007.

[5] J. J. Craig, *Introduction to Robotics.*, 2nd ed. Addison Wesley, 1989.

[6] D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation planning on constraint manifolds," in *IEEE International Conference on Robotics and Automation (ICRA '09)*, May 2009.

[7] S. Ramamoorthy and B. J. Kuipers, "Trajectory generation for dynamic bipedal walking through qualitative model based manifold learning," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 359–366, May 2008.