Supervised Learning for Controlling Fluids

Zherong Pan & Dinesh Manocha*
Department of Computer Science
the University of North Carolina at Chapel Hill

Abstract

We present novel learning-based fluid body controller. The underlying fluid bodies are represented using very high-dimensional configuration spaces, and we use problem-specific dimensionality reduction and controller optimization techniques to manipulate them. These optimized controllers are used for robotic motion planning to manipulate liquids. Specifically, we focus on the problem of liquid transfer where the goal is to move liquid between containers. We show that, when restricted to a specific manipulation problem, the configuration of liquid body lies in a low-dimensional manifold. When this manifold is identified, controller optimization becomes feasible and the optimized controller is computationally efficient.

1 Introduction

Fluid bodies, such as the air we breathe and the water we drink, are ubiquitous in our daily lives. Therefore, the theoretical and computational study of these deformable bodies have drawn considerable attention. Current numerical solvers can simulate the dynamics of such fluids with high accuracy. However, since fluid bodies can undergo arbitrarily large deformations, they are discretized with a large number of elements, and thereby resulting in a high-DOF configuration space representation. Popular discretization methods use either a large set of particles or a high resolution mesh (see [1, 2]). This high-dimensionality results in high computational overhead. For example, current fluid simulation algorithms [3] can take more than 5 hours to model a pool of water with 1.7 million particles on a desktop machine for visual simulation. A highly accurate simulation of dam-break can take hundreds of hours on a small cluster, as reported in [4].

These recent advances in efficient fluid simulation has resulted in the development of different techniques for fluid control. Fluid control problems have many applications in chemistry for reaction control [5], in robotics for liquid manipulation [6], and in computer graphics for physically based animation [7]. However, even though the numerical simulation of fluid bodies is computationally feasible, direct application of tools from optimal control theory to fluid bodies is still regarded as computationally very challenging. For example, iterative dynamic programming [8], a widely used tool in robotics to control nonlinear dynamic systems, has a cost that is at least quadratic in the dimension of configuration space. On the other hand, general-purpose policy search methods [9] require sampling in the configuration space and suffer from the-curse-of-dimensionality.

We summarize our work on the design of novel problem-specific controllers for high dimensional fluid bodies using learning methods. In Section 2, we introduce a system identification method to learn a low-dimensional liquid dynamic model from high-dimensional groundtruth fluid simulation data, which is used for liquid pouring using a robot manipulator. In Section 3, we introduce a new dataset and a neural-net feedback controller for liquid transfer planning.

* {zherong,dm}@cs.unc.edu

NIPS Workshop on Intuitive Physics (NIPS 2016), Barcelona, Spain.

2 Liquid System Identification

In order to design an efficient algorithm for liquid manipulation [10], we try to restrict the degree-of-freedoms (DOFs) of the liquid body to a low-dimensional manifold embedded in its high-dimensional configuration space. The observation behind this idea is that DOFs of a liquid body are mostly used for secondary dynamics (such as splashes, small-scale curls, and turbulences) instead of primary dynamics (i.e. the large-scale laminar flow) that is the prime application of the controller. However, the shape of this manifold is hard to model and scenario specific. As a result, we mainly focus on the problem of liquid transfer, where the goal is to transfer the liquid body between the containers. This is a very important liquid manipulation problem that frequently arises in different robot applications.

Instead of using general-purpose methods in machine learning for manifold identification, we assume that the dynamics of main flow during liquid transfer can be captured by a two-dimensional manifold. Specifically, we use two features inspired by simple rules governing fluid motion, the Bernoulli's equation [11], and rigid body formulation (see Figure 1 (b)). These features depend on only two global variables, as illustrated in Figure 1 (a). The linear coefficients of these features are learned by solving a regression to match the dynamic behaviour with a groundtruth fluid simulation dataset of 127 random simulations.

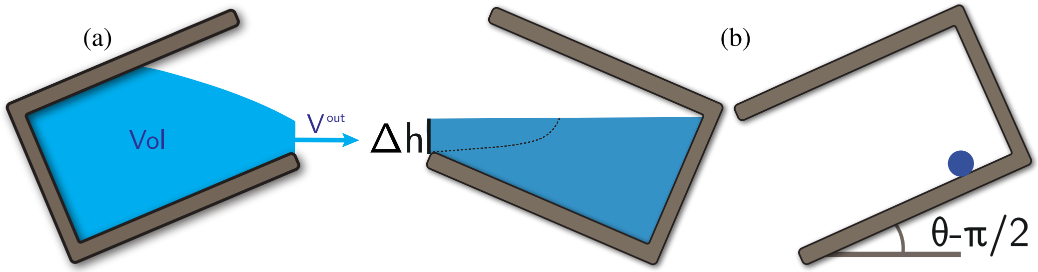

Figure 1: (a): Instead of using high-dimensional representation (e.g., a set of particles), we use only the global variables: V^{out} which is the main outflow velocity, and Vol which is the total volumne of liquid inside the bounding box of source container. (b): The two physically inspired features. Left: If the Bernoulli equation is used between two end points of the dashed streamline, we have $V^{out} \approx \sqrt{2g\Delta h}$. Right: If a single fluid particle sliding down the wall of the container, we have $V^{out} \approx \sin(\theta - \pi/2)$.

After learning the reduced dynamic model, we can perform time integration in the resulting two-dimensional configuration space, which is orders of magnitude faster than accurate fluid simulation. To highlight the benefit, we plugged our model into an optimization based robot motion planner and generate a set of successful liquid transfer trajectories. The computation of each motion plan can be accomplished within 10 minutes compared with more than 3 hours using accurate fluid modelling.

3 Neural-Net Feedback Planner

In Section 2, we proposed a learned liquid dynamic model that is used for motion planning. However, the planner itself uses a conventional optimization-based formulation. In our follow-up work [12], we unify the functionality of liquid dynamics and liquid transfer control using a single neural-net. Trained using carefully prepared data, the neural-net can make decision at real-time rates, and thereby facilitate feedback motion planning.

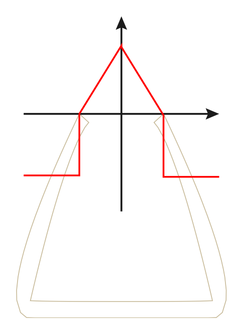

Figure 2: An illustration of our reward function (red). The function is centered around the opening of target container (light gray). We impose higher reward for particles closer to the center of opening and a negative reward for spilled particles.

The preparation of training dataset is crucial to the success of this method. If we are learning for liquid system identification, our training dataset can be obtained from arbitrary liquid transfer simulations, as shown in Section 2. As long as the simulator can generate accurate results, our reduced model can learn the liquid dynamics. However, we are learning not only the liquid dynamics but also the control policy. Therefore, our new training dataset must contain only successful liquid transfer trajectories, where liquid materials finally fall inside our target container.

To generate such trajectories, we use stochastic optimizer (the CMA-ES algorithm) to maximize a reward function in a carefully-designed, problem-specific, low-dimensional search space. As

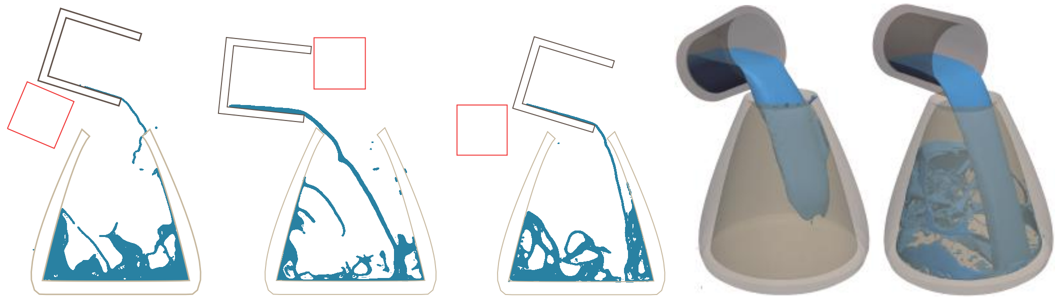

Figure 4: Our real-time controller can be used on new planning problems, such as dynamic obstacle (left) and 3D workspaces (right), although our dataset lives in 2D workspaces.

illustrated in Figure 2, our reward function encourages every discrete liquid material to fall inside the target container by penalizing the distance between the particle's position and center of target container opening. In order to avoid spilling, we further impose a high negative reward for each particle that falls outside the target container. To reduce the dimension of our search space, we use spline interpolation to parameterize the transfer trajectory.

Each such optimization is computationally very costly since we need to compare qualities of many tentative solutions where each comparison requires a fluid re-simulation. In [12], we generated two datasets, TRANSFER and TRANSFER+SPILL. In each dataset, we optimized for a set of 1000 successful pouring trajectories in 2D, where each optimization requires 6000 fluid re-simulations. Altogether, the 6 million fluid re-simulations took 8.1 days of computation on a 1000-core cluster. These trajectories differ in the initial relative position of source and target container, the moving speed of target container, and the amount of liquid in the source container. Moreover, the trajectories in the two datasets differ in the initial liquid velocity. In TRANSFER dataset, we set the initial liquid's velocity to be the same as that of the source container, while in TRANSFER+SPILL dataset, we always set the initial liquid 's velocity to be zero. Therefore, it is more likely that liquids will spill from the source container in TRANSFER+SPILL dataset due to the difference between liquid's velocity and container's velocity (see Figure 3). In order to maximize the reward, trajectories in TRANSFER+SPILL dataset have to learn to avoid spilling.

Using these two datasets, we train a small neural-net that takes as input the observed liquid free surface location and outputs the predicted liquid outflow velocity as well as the desired linear and angular speed of source container for successful liquid transfer (see Figure 3). Note that by using only the sampled free surface as input, we have greatly reduced the dimensional of source container by ignoring all the velocity information. We recover the velocity information by having the neural-net memorize a short history of observed free surface locations.

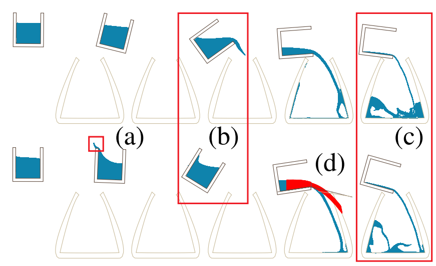

Figure 3: We generated two datasets: TRANSFER and TRANSFER+SPILL. (a): TRANSFER+SPILL encourages spilling by setting initial liquid velocity to zero. (b): The trajectories in TRANSFER+SPILL learns to turn the source container slowly to avoid spilling. (c): Finally, trajectories in both datasets can successfully get all liquids in target container. (d): we sample free surface (red) as a heightfield and feed this to our neural-net.

The neural-net controller trained using this dataset gains desired skills of liquid transfer and more than 85% of discrete liquid elements finally lie inside the target container in all our benchmarks. When we train the Neural-Net using TRANSFER+SPILL dataset, it also learns to avoid spilling by turning the source container slowly at an early stage of liquid pouring. The learned controller can be used in new scenarios not seen in the training dataset, such as scenarios with new fluid materials and 3D workspaces, as illustrated in Figure 4.

4 Limitation and Future Work

A major limitation of our current work is that the controller is problem dependent. For example, we treat free surface as a height field in Section 3. However, this representation excludes more general topology changes such as breaking wave. And the method is limited to laminar flow. These constraints can be relaxed in the future as illustrated in our follow up work [13] where we propose a spacetime optimization technique for keyframe-based smoke control, which can be used as a building block for general reinforcement learning (see [14]). Moreover, as noted by [14], a critical problem in policy search is that the neural-net should be able to represent the dataset. Several methods have been proposed to adapt the dataset based on the state distribution generated by following the trained neural-net. However, updating the dataset is very costly in our case because each update requires several passes of fluid re-simulation which will at least take hours on current machine.

Acknowledgement

This research is supported in part by ARO Contract W911NF-14-1-0437, NSF award 1305286.

References

[1] Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.

[2] Francis H Harlow, J Eddie Welch, et al. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids*, 8(12):2182, 1965.

[3] Ryoichi Ando, Nils Thürey, and Chris Wojtan. Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph. (Proc. SIGGRAPH 2013)*, July 2013.

[4] Timothy R. Langlois, Changxi Zheng, and Doug L. James. Toward animating water with complex acoustic bubbles. *ACM Transactions on Graphics (SIGGRAPH 2016)*, 35(4), July 2016.

[5] Alfio Borzi and R Griesse. Experiences with a space–time multigrid method for the optimal control of a chemical turbulence model. *International journal for numerical methods in fluids*, 47(8-9):879–885, 2005.

[6] Chris Bowen and Ron Alterovitz. Closed-loop global motion planning for reactive execution of learned tasks. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1754–1760. IEEE, 2014.

[7] Adrien Treuille, Antoine McNamara, Zoran Popović, and Jos Stam. Keyframe control of smoke simulations. *ACM Transactions on Graphics*, 22(3):716–723, July 2003.

[8] Tom Erez, Yuval Tassa, and Emanuel Todorov. Infinite-horizon model predictive control for periodic tasks with contacts. *Robotics: Science and systems VII*, page 73, 2012.

[9] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225. IEEE, 2006.

[10] Zherong Pan and Dinesh Manocha. Motion planning for fluid manipulation using simplified dynamics. *arXiv preprint arXiv:1603.02347*, 2016.

[11] Horace Lamb. *Hydrodynamics*. Cambridge university press, 1932.

[12] Zherong Pan and Dinesh Manocha. Feedback motion planning for liquid transfer using supervised learning. *arXiv:1609.03433*, 2016.

[13] Zherong Pan and Dinesh Manocha. Efficient optimal control of smoke using spacetime multigrid. *arXiv:1608.01102*, 2016.

[14] Sergey Levine and Vladlen Koltun. Learning complex neural network policies with trajectory optimization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 829–837, 2014.