Learning Fluid-Directed Rigid Body Control

Kārlis Freivalds, Oskars Teikmanis, Laura Leja, Rodions Saltanovs, Ralfs Āboliņš
Institute of Electronics and Computer Science
Latvia
{karlis.freivalds, oskars.teikmanis, laura.leja,
rodions.saltanovs, ralfs.abolins}@edi.lv

Abstract

We introduce a method for contactless manipulation of objects in three-dimensional space using controlled fluid streams. To achieve this, we train a neural network controller in a differentiable simulation and evaluate it in a simulated environment consisting of a grid of vertical emitters. By carrying out various horizontal displacement tasks such as moving objects to specific positions while reacting to external perturbations, we demonstrate that a controller, trained with a limited number of iterations, can generalise to longer episodes and learn the behavior of fluid-solid interactions. We are in the process of building a floor-standing device that embodies the proposed method, and we describe its key components and design decisions towards displacing and manipulating delicate objects on the device's surface.

1 Introduction

Contactless object manipulation has historically been limited to very small objects, as in the case of acoustic levitation (Chen et al., 2019), or to objects with special properties such as magnetism (Xie et al., 2021). Furthermore, while there has been some research pushing the boundaries of air-based displacement of objects (Iwaki et al., 2011; Bhardwaj et al., 2024), much of the technology remains similar to simple "air bearings," like the ones used in air-hockey tables (Laurent and Moon, 2015).

Real-time control of systems involving fluids and solids present inherent challenges due to the complex dynamics arising from their interactions and high-dimensional state space. The nonlinear nature of these interactions make the the application of classic control approaches, such as PID, highly impractical. In this paper, we overcome this difficulty by leveraging a differentiable physics simulator in conjunction with deep learning techniques. The majority of the work in the field of fluid-directed object manipulation exists in purely simulated environments (Ramos et al., 2022; Freivalds et al., 2024). A prominent work by Ma et al. (2018) uses reinforcement learning (RL) to train movable fluid spouts to interact with rigid objects in a two-dimensional environment. Recently, Google DeepMind published their work on fluid-directed rigid body control on a bench-top device called "Box o' Flows" (Bhardwaj et al., 2024). It showcases the potential of RL in implementing fluid-based control by using an array of spouts that can perform complex contactless manipulations with rigid objects in a 3D environment.

Our approach for training control policies is based on differentiable physics (DP) simulations. Advances in DP (Thuerey et al., 2021) have demonstrated significant potential for addressing simulated inverse problems and complex control applications (Hu et al., 2019; Holl et al., 2020; Du et al., 2021; Fang et al., 2022; Ramos et al., 2022; Teikmanis et al., 2023; Tathawadekar et al., 2023). In the context of computational fluid-solid interactions, there are several examples of applied DP (Ramos et al., 2022; Ma et al., 2021; Hu et al., 2019; Du et al., 2021; Li et al., 2023). However, to our knowledge, there is no existing research where DP is applied for controlling real devices in which the fluid serves as the acting agent.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

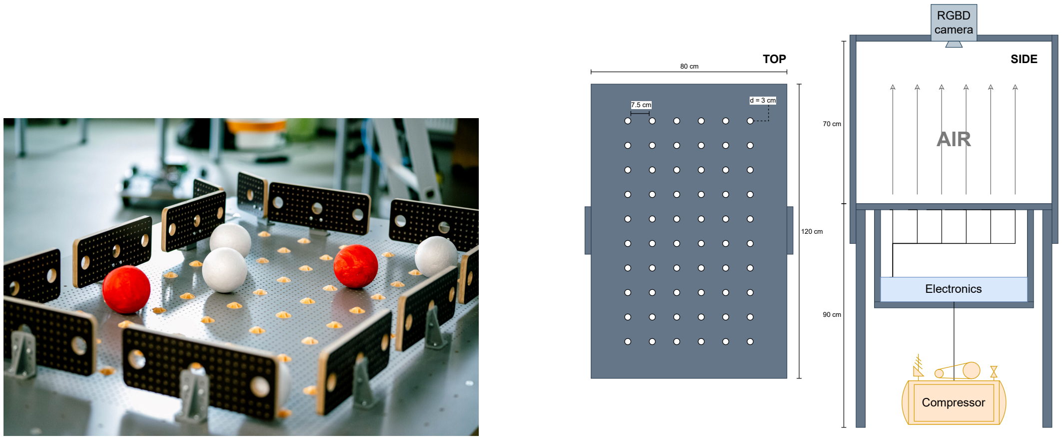


Figure 1: A device prototype for air-based manipulation of rigid objects (left), and its schematic design (right). The device consists of a tabletop array of individually controllable nozzles that output compressed air. An RGBD camera is placed above the table to observe the movement. This information is used by a neural network controller to regulate the fluid emitters to complete various displacement tasks.

In this article, we describe our work on a device designed for fluid-directed manipulation of objects, shown in Fig. 1. It consists of upward facing nozzles arranged in a grid. The output pressure of the entire grid is controllable by a neural network controller, and each nozzle can be opened and closed separately. The objective of this research is to provide a proof-of-concept for diverse, three-dimensional object manipulation with air flows, using machine learning methods that leverage differentiable simulations. We demonstrate that the controller can be successfully trained to move objects across the table.

2 Simulation

The simulation domain is designed as a cuboid with all boundaries open, except for the bottom boundary, which functions as a solid surface. On this surface, emitters are arranged in a grid, each capable of emitting fluid at different velocities. Within this setup, one or more rigid objects are placed to demonstrate controlled fluid-solid interactions. Section 4 features examples with one and two spherical objects with a radius of 10 units, but the approach can handle any number of such objects.

The system behaviour is modelled using fluid dynamics within a bounded domain, and is governed by the incompressible Navier-Stokes (NS) equations and is implemented using the differentiable PDE solver and simulation toolkit PhiFlow (Holl et al., 2020). It supports fluid simulations and one-way coupling with solids, i.e., solid objects influence the fluid but not vice-versa. To enable full coupling, we employ the implementation from Ramos et al. (Ramos et al., 2022), adapted for 3D simulations. With this approach, we can calculate the force exerted by the fluid on solids. For the current implementation, we simulate only spherical objects, for which we implement Newtonian motion and collision handling using PhiFlow's integrated solver.

The domain is discretized at a $32 \times 32 \times 32$ resolution, which is the maximum for training on one A100 GPU card. The emitters are modelled as boundary conditions on the bottom plane of the domain. Each emitter functions as an inlet of a small area, producing fluid velocities as prescribed by the controller for each time step.

3 Neural Network Controller

The fluid velocities are controlled by a neural network controller, which takes the position and velocity of the objects as inputs, and outputs the required velocities for each emitter to guide the objects to their target locations. In a control scenario, the process depicted in Fig. 2 is applied for each time step.

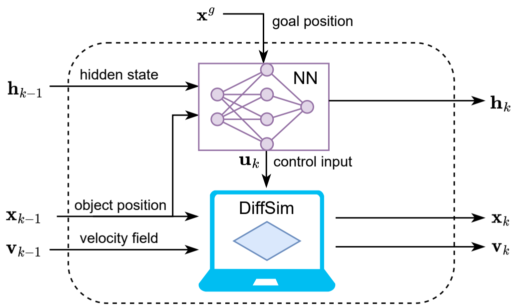


Figure 2: **Differentiable control diagram**. Shown is one time step of the controlled simulation. The differentiable simulator (bottom) operates on the velocity field $\mathbf{v}$ containing fluid velocity vectors for each spatial location at a time step k, the object's 3D position $\mathbf{x}$, and the control signal $\mathbf{u}$, corresponding to the emitter velocity. The neural network controller (top) receives the object's position from the simulator at the previous time step $(k-1)$, its own hidden state $\mathbf{h}$, and the goal position of the controlled objects in order to update its hidden state and the control signal.

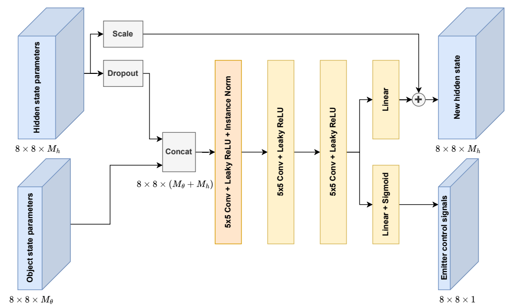


Figure 3: **Neural network architecture**. We use a recurrent convolutional neural network (RCNN) that operates on an input grid of (8×8, in this case) fluid emitters with a corresponding feature map of M_θ elements, and a hidden state of size M_h. Dropout is applied to the hidden state to enhance generalisation to longer simulations. Two output heads performing linear projection are attached to the last layer, producing emitter velocities (range-limited via sigmoid activation) and an updated hidden state for the next time step by applying a scaled residual connection.

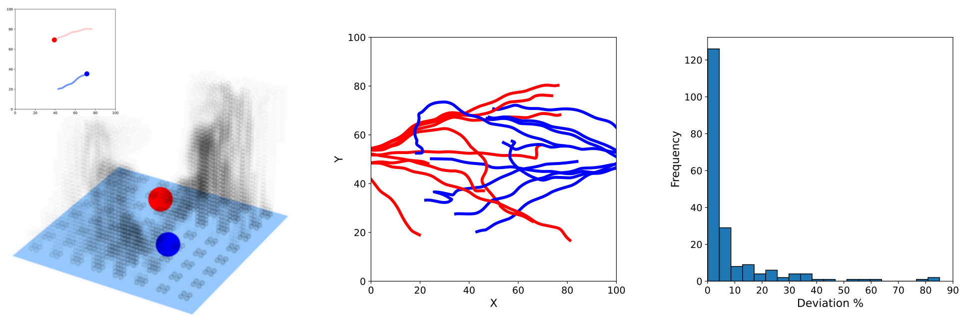
The neural network architecture is depicted in Fig. 3. It is designed as a recurrent convolutional network, receiving the object state representation and its own internal state as inputs at each time step. It produces an emitter control signal matrix and the next internal state tensor as outputs, and operates on spatial data with a shape corresponding to the emitter grid. By design, convolutions are well suited for such spatial data processing. During training, this structure is unrolled through time, and the loss function is calculated depending on $\mathbf{x}_N$, where N is the number of time steps. Gradient based training is performed to adjust the neural network controller's parameters to minimise the loss. Recurrence in the neural network is implemented as scaled residual connections (Zakovskis et al., 2023) similar to ResNet (He et al., 2016). This design together with application of truncated backpropagation through time, where simulation segments of 80 steps are backpropagated, is used to facilitate stable training.

To fit the convolutional structure, the object state is given to the neural network as a matrix of the same size as the emitter grid and several feature maps. In total, there are seven feature maps per controlled object giving the object's position, velocity, and the goal position. To specify the position, the object is rendered onto the matrix using soft boundaries giving the volume fraction occupied by the object, similar to (Takahashi et al., 2021). The goal position $\mathbf{x}^g$ is represented as a distance field. Object velocity is supplied as the difference of two rendered positions where the second one shows the object displaced by its velocity vector. Such representations support gradient backpropagation.

To train the system, we initialise objects with randomised locations and velocities, conduct simulations, and use the object trajectory for loss calculation. The loss function primarily expresses the distance of the object to the desired target location $\mathbf{x}^g$ with additional terms:

$$L = ||x_N - x^g||^2 + \alpha \sum_{k=1}^{N} ||x_k - x^g||^2 + \beta \sum_{k=1}^{N} ||\mathbf{u}_k||^2. \tag{1}$$

The first term ensures that the final object position is close to the target position. The second term keeps all intermediate positions near the target, effectively maximising the object's speed towards the target and maintaining its position once it arrives. We experimentally set $\alpha = 1/N$ to achieve a good balance with the first and most important term. The third term aims to minimise the total energy expended by the emitters throughout the episode. We set a small weight $\beta = 0.001$ for this term during training.



(a) (b) (c)

Figure 4: **Controlled displacement of two simulated objects**. (a) The solid lines in the top left shows the trajectory of the objects as they are moved by the simulated fluid. The fluid is emitted from an 8×8 grid, and is visualised as a grey plume. (b) Sample trajectories of moved two objects, when the red object should reach the point $(-10, 50)$ on the left, and the blue object should $(110, 50)$ on the right. (c) Percentage by which the actual path deviates from a straight line, for 100 trajectories. Animations can be found in the accompanied video.

All steps of the simulation are differentiable, and we employ gradient descent with the Adam optimiser (Kingma and Ba, 2015) using a learning rate of 0.0001. Based on experiments, the training is stable for values between 0.001 and 0.0001, resulting in minor differences in training speed. Training is performed for 2000 iterations, which takes about about 14 hours on a single Nvidia A100 GPU.

4 Controller Evaluation

We evaluate the trained controller in three different scenarios that showcase the capabilities to carry out versatile object displacement using controlled fluid flows: a) hold the object in the given position above the table b) move the object to a desired location, c) move two objects simultaneously to different target locations. We also analyse the system's robustness against external perturbations such as a side current. A horizontal current of a random direction and magnitude (up to 20% of the emitter velocity) is applied during both training and testing. During these evaluations, simulations are conducted over 600 time steps, which is sufficient to accomplish the specified tasks. The training succeeds for all the considered tasks yielding positive results in simulation. The most difficult task is to control two objects simultaneously. The results of this task are depicted in Fig. 4 where we see that the controller has learnt to correctly move both objects to their respective targets in relatively straight trajectories except in cases when both objects come into close proximity. In these cases, the emitters are able to separate them successfully, as demonstrated in the video[1].

5 Conclusion

We have given an overview of our ongoing work on a floor-standing device for contactless manipulation of rigid objects. We have developed a method to train a neural network controller that can perform object displacement tasks using an array of fixed vertical fluid emitters. This approach has demonstrated versatile control tasks across multiple simulated scenarios. Our controller has also proven robust against external disturbances, which is an essential feature for bridging the gap between simulations and real-world devices. The controller, trained by leveraging a differentiable simulation, reaches good performance within just 2000 iterations and can handle much longer episodes than those used during training. A limitation of the current implementation is that it can only handle spherical objects. To support arbitrary objects, more advanced collision detection and fluid-rigid coupling methods need to be incorporated. In future work, we will develop sim-to-real strategies to implement the described control scenarios on the physical device.

[1]https://www.youtube.com/watch?v=argalwINGwc

Acknowledgments and Disclosure of Funding

This research is funded by the Latvian Council of Science, project "Smart Materials, Photonics, Technologies and Engineering Ecosystem", project No. VPP-EM-FOTONIKA-2022/1-0001.

References

M. Bhardwaj, T. Lampe, M. Neunert, F. Romano, A. Abdolmaleki, A. Byravan, M. Wulfmeier, M. Riedmiller, and J. Buchli. Real-World Fluid Directed Rigid Body Control via Deep Reinforcement Learning. *arXiv preprint arXiv:2402.06102*, 2024.

X. Chen, K. H. Lam, R. Chen, Z. Chen, X. Qian, J. Zhang, P. Yu, and Q. Zhou. Acoustic levitation and manipulation by a high-frequency focused ring ultrasonic transducer. *Applied Physics Letters*, 114(5), 2019.

T. Du, K. Wu, P. Ma, S. Wah, A. Spielberg, D. Rus, and W. Matusik. Diffpd: Differentiable projective dynamics. *ACM Trans. Graph.*, 41(2), 11 2021. ISSN 0730-0301. doi: 10.1145/3490168. URL `https://doi.org/10.1145/3490168`.

Y. Fang, J. Liu, M. Zhang, J. Zhang, Y. Ma, M. Li, Y. Hu, C. Jiang, and T. Liu. Complex locomotion skill learning via differentiable physics. *arXiv preprint arXiv:2206.02341*, 2022.

K. Freivalds, L. Leja, and O. Teikmanis. Learning to Move Objects with Fluid Streams in a Differentiable Simulation. *arXiv preprint arXiv:2404.18181*, 2024.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

P. Holl, V. Koltun, and N. Thuerey. Learning to Control PDEs with Differentiable Physics. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand. DiffTaichi: Differentiable Programming for Physical Simulation. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

S. Iwaki, H. Morimasa, T. Noritsugu, and M. Kobayashi. Contactless manipulation of an object on a plane surface using multiple air jets. In *2011 IEEE International Conference on Robotics and Automation*, pages 3257–3262. IEEE, 2011.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

G. J. Laurent and H. Moon. A survey of non-prehensile pneumatic manipulation surfaces: principles, models and control. *Intelligent Service Robotics*, 8:151–163, 2015.

Z. Li, Q. Xu, X. Ye, B. Ren, and L. Liu. Difffr: Differentiable sph-based fluid-rigid coupling for rigid body control. *ACM Transactions on Graphics (TOG)*, 42(6):1–17, 2023.

P. Ma, Y. Tian, Z. Pan, B. Ren, and D. Manocha. Fluid directed rigid body control using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

P. Ma, T. Du, J. Z. Zhang, K. Wu, A. Spielberg, R. K. Katzschmann, and W. Matusik. Diffaqua: A differentiable computational design pipeline for soft underwater swimmers with shape interpolation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

B. Ramos, F. Trost, and N. Thuerey. Control of Two-way Coupled Fluid Systems with Differentiable Solvers. In *ICLR 2022 Workshop on Generalizable Policy Learning in Physical World*, 2022.

T. Takahashi, J. Liang, Y.-L. Qiao, and M. C. Lin. Differentiable fluids with solid coupling for learning and control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6138–6146, 2021.

N. N. Tathawadekar, N. A. K. Doan, C. F. Silva, and N. Thuerey. Incomplete to complete multiphysics forecasting: a hybrid approach for learning unknown phenomena. *Data-Centric Engineering*, 4: e27, 2023.

O. Teikmanis, L. Leja, and K. Freivalds. Applying a Differentiable Physics Simulation to Move Objects with Fluid Streams. In *International Workshop on Embedded Digital Intelligence (IWoEDI)*, 2023.

N. Thuerey, P. Holl, M. Mueller, P. Schnell, F. Trost, and K. Um. *Physics-based Deep Learning*. WWW, 2021. URL `https://physicsbaseddeeplearning.org`.

J. Xie, P. Zhao, C. Zhang, J. Fu, and L.-S. Turng. Current state of magnetic levitation and its applications in polymers: A review. *Sensors and Actuators B: Chemical*, 333:129533, 2021.

R. Zakovskis, A. Draguns, E. Gaile, E. Ozolins, and K. Freivalds. Gates are not what you need in rnns. In *International Conference on Artificial Intelligence and Soft Computing*, pages 304–324. Springer, 2023.