



OptiTrap: Optimal Trap Trajectories for Acoustic Levitation Displays

VIKTORIJA PANEVA and ARTHUR FLEIG, University of Bayreuth

DIEGO MARTÍNEZ PLASENCIA, University College London

TIMM FAULWASSER, TU Dortmund University

JÖRG MÜLLER, University of Bayreuth

Acoustic levitation has recently demonstrated the ability to create volumetric content by trapping and quickly moving particles along reference paths to reveal shapes in mid-air. However, the problem of specifying physically feasible trap trajectories to display desired shapes remains unsolved. Even if only the final shape is of interest to the content creator, the trap trajectories need to determine where and when the traps need to be, for the particle to reveal the intended shape. We propose *OptiTrap*, the first structured numerical approach to compute trap trajectories for acoustic levitation displays. Our approach generates trap trajectories that are physically feasible and nearly time-optimal, and reveal generic mid-air shapes, given only a reference path (i.e., a shape with no time information). We provide a multi-dimensional model of the acoustic forces around a trap to model the trap-particle system dynamics and compute optimal trap trajectories by formulating and solving a non-linear path following problem. We formulate our approach and evaluate it, demonstrating how *OptiTrap* consistently produces feasible and nearly optimal paths, with increases in size, frequency, and accuracy of the shapes rendered, allowing us to demonstrate larger and more complex shapes than ever shown to date.

CCS Concepts: • Computing methodologies → Computer graphics; Motion path planning; Computational control theory;

Additional Key Words and Phrases: Ultrasonic levitation, minimum time problems, path following, volumetric displays, phased arrays of transducers

ACM Reference format:

Viktorija Paneva, Arthur Fleig, Diego Martínez Plasencia, Timm Faulwasser, and Jörg Müller. 2022. OptiTrap: Optimal Trap Trajectories for Acoustic Levitation Displays. *ACM Trans. Graph.* 41, 5, Article 173 (October 2022), 14 pages.
<https://doi.org/10.1145/3517746>

1 INTRODUCTION

Acoustic levitation has recently demonstrated the creation of volumetric content in mid-air by exploiting the **Persistence of Vision (PoV)** effect. This is achieved by using acoustic traps to rapidly

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 737087 (Levitate) and from the AHRC UK-China Research-Industry Creative Partnerships (AH/T01136X/2). Authors' addresses: V. Paneva, A. Fleig, and J. Müller, University of Bayreuth, Universitätsstraße 30, Bayreuth, 95447, Germany; emails: vpaneva@acm.org, {arthur.fleig, joerg.mueller}@uni-bayreuth.de; D. M. Plasencia, University College London, 169 Euston Road, London, NW1 2AE, United Kingdom; email: d.plasencia@ucl.ac.uk; T. Faulwasser, TU Dortmund University, Emil-Figge-Straße 70, Dortmund, 44227, Germany; email: timm.faulwasser@tudortmund.de.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2022 Copyright held by the owner/author(s).
0730-0301/2022/10-ART173
<https://doi.org/10.1145/3517746>

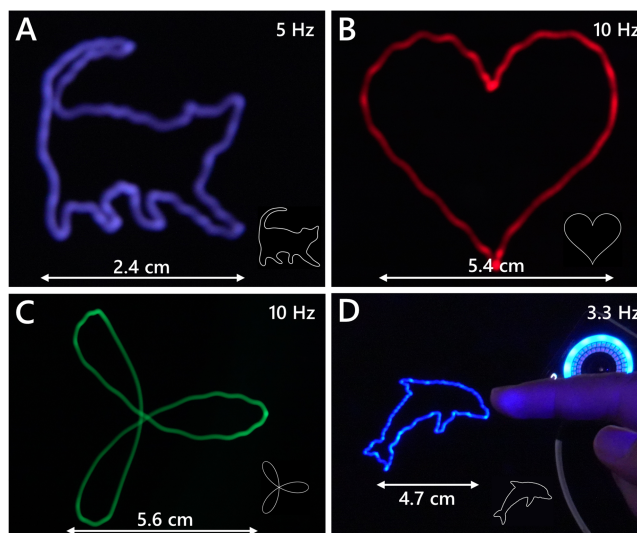


Fig. 1. While previously, shapes demonstrated on levitation displays were limited to simple shapes with almost constant curvature [Fushimi et al. 2019; Hirayama et al. 2019; Plasencia et al. 2020], our approach allows rendering generic complex paths. Shapes that have not been demonstrated before including sharp edges as with the heart (B) and dolphin (D), as well as significant changes of the curvature, as with the cat (A) and flower (C) (see Supplementary Video). Photos of the physical particle are made with an exposure time ranging from 0.2 to 1 s, such that each photo shows multiple periods of the orbit.

move single [Hirayama et al. 2019] or multiple [Plasencia et al. 2020] particles along a periodic reference path, revealing a shape within 0.1s (i.e., the integration interval of human eyes [Bowen et al. 1974]).

However, the way to define such PoV content remains unsolved. That is, there are no automated approaches to compute the *trap trajectories*, i.e., the positioning and timing of the acoustic traps that will allow us to reveal the desired shape.

Currently, content creators can only rely on trial and error to find physically feasible trap trajectories resembling their intended target shapes, resulting in a time-consuming and challenging process. For instance, the creator will need to define the timing of the path (i.e., not only *where* the particle must be, but also *when*). Such timing is not trivial and will affect the overall rendering time and the accelerations applied to the particle, which must be within the capabilities of the levitator. The way forces distribute around the acoustic trap will also need to be considered to decide where traps must be located.

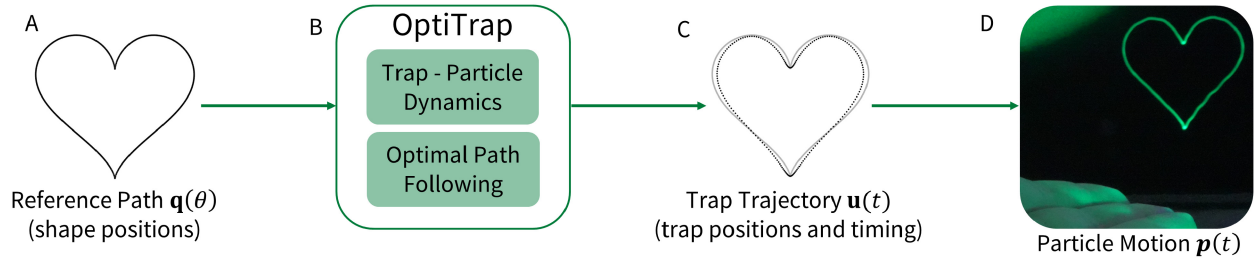


Fig. 2. *OptiTrap* is an automated method to compute trap trajectories to reveal generic mid-air shapes on levitation displays. The method accepts a reference path (e.g., the shape of a heart) without any timing information as an input (A). Our approach considers the capabilities of the device and its trap-particle dynamics and combines these with a path-following approach (B). The approach produces feasible trap trajectories, describing when and where the traps must be created (C). The resulting particle motion can be presented on the actual device, yielding feasible paths and supporting complex objects with sharp edges and/or significant changes of the curvature (D).

Considering these challenges is crucial to design feasible trap trajectories, as a single infeasible point along the trajectory typically results in the particle being ejected from the levitator (e.g., approaching a sharp corner too quickly). Thus, while it has been theorised that linear PoV paths of up to 4 m and peak speeds of 17 m/s are possible [Fushimi et al. 2020], actual content demonstrated to date has been limited to simple shapes with almost constant curvature along the path and much lower speeds (e.g., 0.72 m/s in [Ochiai et al. 2014]; 1.2 m/s in [Hirayama et al. 2019]; 2.25 m/s in [Plasencia et al. 2020], combining six particles).

In this article, we present *OptiTrap*, the first structured numerical approach to compute trap trajectories for acoustic levitation displays. *OptiTrap* automates the definition of levitated PoV content, computing physically feasible and nearly time-optimal trap trajectories given only a reference path, i.e., the desired shape. As shown in Figure 1, this allows for larger and more complex shapes than previously demonstrated, as well as shapes featuring significant changes in curvature and/or sharp corners.

Our approach is summarised in Figure 2. *OptiTrap* assumes only a generic reference path $\mathbf{q}(\theta)$ as an input, with no temporal information (see Figure 2(a)). *OptiTrap* formulates this as a path following problem (see Figure 2(b)), computing the optimum timing in which a particle can traverse such path. Our formulation considers the *Trap-Particle Dynamics* of the system, using a 3D model of acoustic forces around a trap. This results in a non-invertible model, which cannot exploit differential flatness [Fliess et al. 1995], on which most path-following approaches rely. We instead provide a coupling stage for the dynamics of our system and numerically invert the system. This approach produces a trap trajectory $\mathbf{u}(t)$ (Figure 2(c)) that results in the intended and nearly time-optimal particle motion. That is, $\mathbf{u}(t)$ defines the positions and timing of the traps that cause the particle to reveal the target shape $\mathbf{q}(\theta)$, according to the capabilities of the actual device, cf. Figure 2(d).

In summary, we contribute the first structured numerical approach to compute physically feasible and nearly time-optimal trap trajectories for levitation displays, given only a reference path that the particle should follow. As a core contribution, we provide a theoretical formulation of the problem in terms of path following approaches, allowing optimum timing and device properties (e.g., trap-particle dynamics) to be jointly considered. All the details of the approach we propose are provided in Section 4.

We illustrate the potential of our approach by rendering shapes featuring straight lines, sharp corners, and complex shapes, such as those in Figure 1. We then provide an experimental validation of our approach, demonstrating increases of up to 563% in the size of rendered objects, up to 150% in the rendering frequency, and improvements in accuracy (e.g., shape revealed more accurately). While the baseline shapes we compare against require trial and error to determine optimal sizes or frequencies, our approach always yields feasible paths (i.e., working in at least 9 out of 10 attempts) and makes consistent use of accelerations very close to (but not exceeding) the maximum achievable by the device, independently of the target shape.

These features allow *OptiTrap* to render complex objects involving sharp edges and significant changes in curvature that have never been demonstrated before. Even more importantly, it provides a tool to systematically explore the range of contents that levitation displays can create, as a key step to exploit their potential.

2 BACKGROUND AND CHALLENGES

Our goal is to minimise complexity for content creators. Thus, given a reference path (i.e., a shape defined by the content creator), our approach must produce physically feasible trap trajectories, which accurately reveal the desired path, while making optimal use of the capabilities of the device.

We formalise our content as a reference path Q , provided by the content creator without any preassigned time information. It is an explicitly parametrized curve

$$Q := \{\xi \in \mathbb{R}^3 \mid \theta \in [\theta_0, \theta_f] \mapsto \mathbf{q}(\theta)\}, \quad (1)$$

where $\mathbf{q}: \mathbb{R} \rightarrow \mathbb{R}^3$, as later justified in Section 4.2, must be a twice continuously differentiable function with respect to θ , which is called the path parameter. Increasing values of θ denote forward movement along the path Q . The starting point on the path is $\mathbf{q}(\theta_0)$, while $\mathbf{q}(\theta_f)$ marks the end of the path. For periodic paths, such as a particle cyclically revealing a PoV shape, $\mathbf{q}(\theta_0) = \mathbf{q}(\theta_f)$ and $\dot{\mathbf{q}}(\theta_0) = \dot{\mathbf{q}}(\theta_f)$ hold. Please note that while splines would provide a generic solution to this parametrization (i.e., twice continuously differentiable fitting the points in Q), any other parametric functions can be used. For example, taking $\theta \in [0, 2\pi]$ and $r > 0$, a

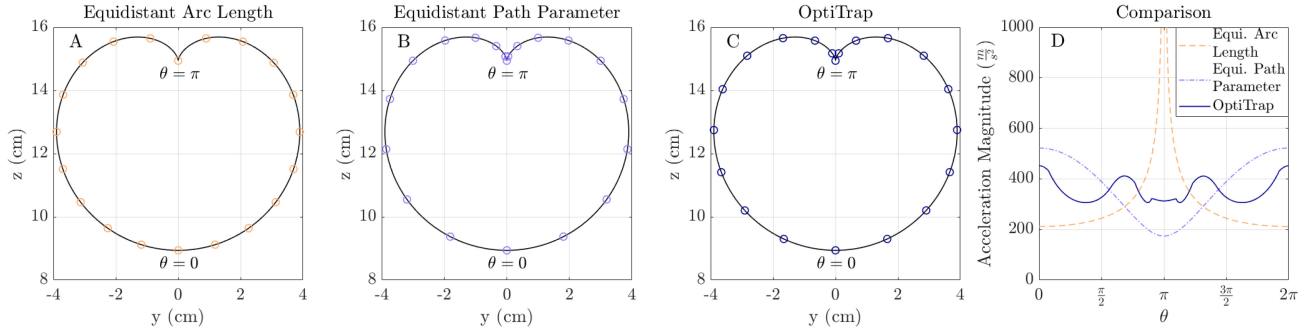


Fig. 3. Three different path timings for the cardioid shape with fixed traversal time: (A) equidistant sampling of the arc length, (B) equidistant sampling of the path parameter θ , and (C) *OptiTrap*. The corresponding acceleration magnitudes are depicted in (D). Strategy (A) is physically infeasible due to the infinite required acceleration at the corner ($\theta = \pi$). Strategy (B) does not share this problem, but requires careful parametrization of the path, while *OptiTrap* (C) yields the timing automatically.

cardioid as in Figure 3, can be described by

$$\mathbf{q}(\theta) = (0, r \sin(\theta)(1 + \cos(\theta)), -r \cos(\theta)(1 + \cos(\theta)) + r)^\top. \quad (2)$$

Whatever the approach used, revealing such reference paths typically involves rapid particle movements, as PoV content must be revealed within about 0.1s [Bowen et al. 1974] and must consider the feasibility of the path (i.e., the trap-particle dynamics, the topology of the trap, and the capabilities of the levitator). This results in the following two main challenges for *OptiTrap*: (1) determining the optimal timing for the particle revealing the path; and (2) computing the trap positions generating the required forces on the particle.

2.1 Challenge 1: Determining an Optimal Path Timing

The reference path Q defines the geometry but not the timing (i.e., it describes *where* the particle needs to be, but not *when* it needs to be there). Our approach must compute such a timing, and the strategy followed will have important implications on the velocity and accelerations applied to the particle and, hence, on the physical feasibility of the content.

Different timing strategies and their effects on the feasibility can be illustrated using the cardioid example from Equation (2). Figure 3 depicts various timing strategies applied to this shape, all of them traversing the same path in the same overall time.

A straightforward approach would be to sample equidistantly along the path, as shown in Figure 3(a). This works well for lines and circles, where a constant speed can be maintained, but fails at sudden changes in curvature due to uncontrolled accelerations (see the acceleration at the corner of cardioid in Figure 3(d), at $\theta = \pi$).

The second example in Figure 3(b) shows a simple equidistant sampling on the path parameter θ . As shown in Figure 3(d), this results in areas of strong curvature naturally involving low accelerations, and can work particularly well for low-frequency path parametrizations in terms of sinusoids (e.g., [Fushimi et al. 2020] showed such sinusoidal timing along straight paths, to theorise optimum/optimistic content sizes). In any case, the quality of the result obtained by this approach will vary depending on the specific shape and parametrization used.

As an alternative, the third example shows the timing produced by our approach, based on the shape and capabilities of the device (i.e., forces it can produce in each direction). Please note how this

timing reduces the maximum accelerations required (i.e., retains them below the limits of the device), by allowing the particle to travel faster in parts that were unnecessarily slow (e.g., Figure 3(d), at $\theta = \pi$). For the same maximum accelerations (i.e., the same device), this timing strategy could produce larger shapes or render them in shorter times, for better refresh rates. This illustrates the impact of a careful timing strategy when revealing any given reference path. We address this challenge in Section 4.2.

2.2 Challenge 2: Computation of Trap Positions

In addition to the timing, the approach must also compute the location of the traps. Previous approaches [Fushimi et al. 2020; Hirayama et al. 2019; Plasencia et al. 2020] placed the traps along the shape to be presented, under the implied assumption that the particle would remain in the centre of such trap.

However, the location where the traps must be created almost never matches the location of the particle. Acoustic traps feature (almost) null forces at the centre of the trap and high restorative forces around them [Marzo et al. 2015]. As such, the only way a trap can accelerate a particle is by having it placed at a distance from the centre of the trap. Such trap-particle distances were measured by [Hirayama et al. 2019] to assess performance achieved during their speed tests. Distortions related to rendering fast-moving PoV shapes were also shown in [Fushimi et al. 2019]. However, none of them considered or corrected for such displacements.

Please note the specific displacement between the trap and the particle will depend on several factors, such as the trap acceleration required at each point along the shape, as well as the trap topology (i.e., how forces distribute around it). No assumptions can be made that the traps will remain constrained to positions along or tangential to the target shape. We describe how our approach solves this challenge in Section 4.3.

3 RELATED WORK

OptiTrap addresses the challenges above by drawing from advances in the fields of acoustic levitation and control theory, which we review in this section.

3.1 Acoustic Levitation

Single frequency soundwaves were first observed to trap dust particles in the lobes of a standing wave more than 150 years

ago [Stevens 1899]. This has been used to create mid-air displays with particles acting as 3D voxels [Ochiai et al. 2014; Omirou et al. 2015, 2016; Sahoo et al. 2016], but such standing waves do not allow control of individual particles. Other approaches have included Bessel beams [Norasikin et al. 2019], self-bending beams [Norasikin et al. 2018], boundary holograms [Inoue et al. 2019] or near-field levitation [Andrade et al. 2016].

However, most display approaches have relied on the generic levitation framework proposed by [Marzo et al. 2015], combining a focus pattern and a levitation signature. Although several trap topologies (i.e., twin traps, vortex traps, or bottle beams) and layouts (i.e., one-sided, two-sided, v-shape) are possible, most displays proposed have adopted a top-bottom levitation setup and twin traps, as these result in highest vertical trapping forces. This has allowed individually controllable particles/voxels [Marzo and Drinkwater 2019] or even particles attached to other props and projection surfaces for richer types of content [Fender et al. 2021; Morales et al. 2019]. The use of single [Fushimi et al. 2019; Hirayama et al. 2019] or multiple [Plasencia et al. 2020] fast-moving particles have allowed for dynamic and free-form volumetric content, but is still limited to small sizes and simple vector graphics [Fushimi et al. 2020].

Several practical aspects have been explored around such displays, such as selection [Freeman et al. 2018] and manipulation techniques [Bachynskiy et al. 2018], content detection and initialisation [Fender et al. 2021] or collision avoidance [Reynal et al. 2020].

However, no efforts have been made towards optimising content considering the capabilities (i.e., the dynamics) of such displays, particularly for challenging content such as the one created by PoV high-speed particles. [Hirayama et al. 2019] showed sound-fields must be updated at very high rates for the trap-particle system to engage in high accelerations, identifying optimum control for rates above 10 kHz. However, they only provided a few guidelines (e.g., maximum speed in corners, maximum horizontal/vertical accelerations) to guide the definition of the PoV content. [Fushimi et al. 2020] provided a theoretical exploration of this topic, looking at maximum achievable speeds and content sizes, according to the particle sizes and sound frequency used. While the dynamics of the system were considered, these were extremely simplified, using a model of acoustic trap forces and system dynamics that are only applicable for oscillating recti-linear trajectories along the vertical axis of the levitator. [Paneva et al. 2020] proposed a generic system simulating the dynamics of a top-bottom levitation system. This operates as a forward model simulating the behaviour of the particle given a specific path for the traps, but unlike our approach, it does not address the inverse problem. Thus, *OptiTrap* is the first algorithm allowing the definition of PoV content of generic shapes, starting only from a geometric definition (i.e., shape to present, no timing information) and optimising it according to the capabilities of the device and the dynamics of the trap-particle system.

3.2 Path Following and Optimal Control

The particle in the trap constitutes a dynamical system, which can be controlled by setting the location of the trap. As such, making the particle traverse the reference path can be seen as an **optimal control problem (OCP)**.

In the context of computer graphics, optimal control approaches have been studied particularly in the areas of physics-based character animation [Geijtenbeek and Pronost 2012] and aerial videography [Nägeli et al. 2017]. Such OCPs can be considered as function-space variants of nonlinear programs, whereby nonlinear dynamics are considered as equality constraints. In engineering applications, OCPs are frequently solved via direct discretization [Bock and Plitt 1984; Von Stryk 1993], which leads to finite-dimensional nonlinear programs. In physics-based character animation, such direct solution methods are known as spacetime constraints [Rose et al. 1996; Witkin and Kass 1988].

If framed as a *levitated particle along a given geometric reference path* (i.e., the PoV shape), the problem can be seen as an instance of a path following problem [Faulwasser 2012]. Such problems also occur in aerial videography, when a drone is to fly along a reference path [Nägeli et al. 2017; Roberts and Hanrahan 2016]. In order to yield a physically feasible trajectory, it is necessary to adjust the timing along that trajectory. This can be done by computing feed-forward input signals [Faulwasser 2012; Roberts and Hanrahan 2016] or, if the system dynamics and computational resources allow, using closed-loop solutions, such as Model Predictive Control in [Nägeli et al. 2017].

However, optimal control of acoustically levitated particles cannot be approached using such techniques. All of the above approaches require *differential flatness* [Fliess et al. 1995]. That is, the underlying system dynamics must be invertible, allowing for the problem to be solved by projecting the dynamics of the moving object onto the path manifold [Nielsen and Maggiore 2008]. This inversion approach also enables formulating general path-following problems in the language of optimal control, encoding desired objectives (e.g., minimum time, see [Faulwasser et al. 2014; Shin and McKay 1985; Verscheure 2009]).

As discussed later in Section 4.1, the distribution of forces around the acoustic trap (i.e., our system dynamics) are not invertible, requiring approaches that have not been widely developed in the literature. We do this by coupling the non-invertible particle dynamics into a virtual system, solvable with a conventional path following approach. We then use these coupling parameters and our model of particle dynamics to solve for the location of the traps.

4 OPTIMAL CONTROL FOR LEVITATION DISPLAYS

This section provides a description of our *OptiTrap* approach, which automates the definition of levitated content, computing physically feasible and nearly time-optimal trap trajectories using only a reference path as an input. In Section 4.1, we first describe our specific hardware setup and the general mathematical framework, then we consider existing models of the trap-particle dynamics (Section 4.1.1), before introducing our proposed model (Section 4.1.2). Next, we describe the two stages in our algorithm, which match the challenges identified in Section 2. That is, Section 4.2 describes the computation of optimum timing, approached as an optimal path following problem, while Section 4.3 explains how to compute the trap locations. Please note that Section 4 focuses on the general case of presenting levitated content, that is, particles cyclically traversing a reference path (shape) as to reveal it. Other cases, such as a particle accelerating from rest

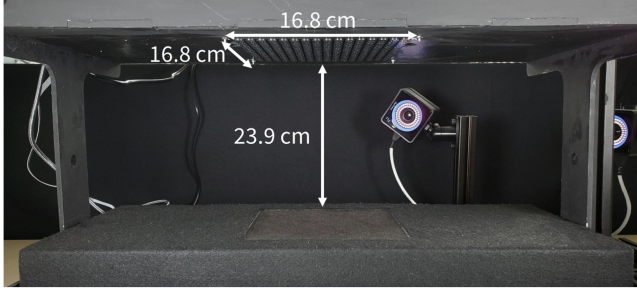


Fig. 4. Overview of the components in our setup. We used two opposed arrays of transducers at a distance of 23.9 cm and an OptiTrack system to track the position of levitated particles in real-time.

as to reach the initial state (i.e., initial position and speed) required to render the content can be easily derived from the general case presented here, and are detailed in the Supplementary Material S1.

4.1 Modelling the Trap-Particle Dynamics

We start by describing the model of the trap-particle dynamics used for our specific setup, shown in Figure 4. This setup uses two opposed arrays of 16×16 transducers controlled by an FPGA and an OptiTrack tracking system (Prime 13 motion capture system at a frequency of 240 Hz). The design of the arrays is a reproduction of the setup in [Morales et al. 2021], modified to operate at 20 Vpp and higher update rates of 10 kHz. The device generates a single twin-trap using the method described in [Hirayama et al. 2019], allowing for vertical and horizontal forces of $4.2 \cdot 10^{-5}$ N and $2.1 \cdot 10^{-5}$ N, respectively, experimentally computed using the linear speed tests in [Hirayama et al. 2019] (i.e., 10cm paths, binary search with 9 out of 10 success ratios, with particle mass $m \approx 0.7 \cdot 10^{-7}$ kg). Note the substantial difference between the maximum forces in the vertical and horizontal directions. Our approach will need to remain aware of direction as this determines maximum accelerations.

In general, the trap-particle dynamics of such a system can be described by simple Newtonian mechanics, i.e.,

$$m\ddot{\mathbf{p}}(t) = \mathbf{F}(\mathbf{p}(t), \dot{\mathbf{p}}(t), \mathbf{u}(t)). \quad (3)$$

Here, $\mathbf{p}(t) = (p_x, p_y, p_z)^\top \in \mathbb{R}^3$ represents the particle position in Cartesian (x, y, z) coordinates at time $t \in \mathbb{R}_0^+$, and $\dot{\mathbf{p}}(t)$ and $\ddot{\mathbf{p}}(t)$ are the velocity and acceleration of the particle, respectively. The force acting on the particle is mostly driven by the acoustic radiation forces and, as such, drag and gravitational forces can be neglected [Hirayama et al. 2019]. Therefore, the net force acting on the particle will depend only on $\mathbf{p}(t)$ and on the position of the acoustic trap at time t denoted by $\mathbf{u}(t) = (u_x(t), u_y(t), u_z(t))^\top \in \mathbb{R}^3$.

As a result, our approach requires an accurate and ideally invertible model of the acoustic forces delivered by an acoustic trap. That is, we seek an accurate model predicting forces at any point around a trap only in terms of $\mathbf{u}(t)$ and $\mathbf{p}(t)$. Moreover, an invertible model would allow us to analytically determine where to place the trap as to produce a specific force on the particle given the particle location.

For spherical particles considerably smaller than the acoustic wavelength and operating in the far-field regime, such as those

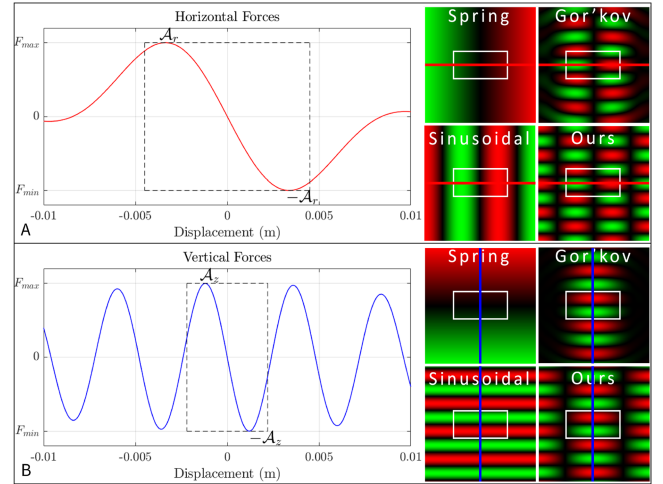


Fig. 5. Analytical models of horizontal (A) and vertical (B) acoustic forces around a trap. The plots on the left show forces along the main axes X and Z , and analytical models provide a good fit. The plots on the right show horizontal (A) and vertical (B) forces across 2D slices through the trap centre along the X and Z axes. Spring and Sinusoidal models fail to predict forces outside the main axes, while our proposed model provides accurate reconstruction within the region of interest (highlight).

used by our device, the acoustic forces exerted can be modelled by the gradient of the Gor'kov potential [Bruus 2012]. This is a generic model, suitable to model acoustic forces resulting from any combination of transducer locations and transducer activations, but it also depends on all these parameters, making it inadequate for our approach.

Our case, using a top-bottom setup and vertical twin traps is much more specific. This allows for simplified analytical models with forces depending only on the relative position of the trap and the particle, which we compare to forces as predicted from the Gor'kov potential in Figure 5.

4.1.1 Existing Models of the Trap-Particle Dynamics. Simple *spring* models have been used extensively [Fushimi et al. 2020, 2018; Paneva et al. 2020], modelling trapping forces according to a *stiffness* parameter $\mathcal{K}_i$, that is, with forces being proportional to the distance of the particle to the centre of the trap

$$F_i(\mathbf{p}, \mathbf{u}) := \mathcal{K}_i \cdot |u_i - p_i|, \quad i \in \{x, y, z\}.$$

Such models are usually refined by providing specific stiffness values for each dimension, but they are only suitable for particles remaining in close proximity to the centre of the trap, i.e., the linear region near the centre of the trap. As a result, *spring* models are only accurate for systems moving particles slowly, requiring low acceleration and forces (so that particles remain within the linear region).

Sinusoidal models have been proposed as an alternative [Fushimi et al. 2020, 2018], providing accurate fitting from the centre of the trap to the peaks of the force distribution in Figure 5, according to the peak trapping force $\mathcal{A}_i$ and characteristic

frequency $\mathcal{V}_i$:

$$F_i(\mathbf{p}, \mathbf{u}) := \mathcal{A}_i \cdot \sin(\mathcal{V}_i \cdot (u_i - p_i)), \quad i \in \{x, y, z\},$$

However, both of these models (i.e., *spring* and *sinusoidal*) are only suitable for particles placed along one of the main axes of the acoustic trap, not for particles arbitrarily placed at any point around it. This is illustrated on the right of Figure 5, which provides an overview of how forces distribute on a horizontal and vertical 2D plane around a trap, according to each model (i.e., Gor'kov, *spring*, *sinusoidal*, and *Ours*, detailed in the next subsection).

Please note how the three previous models show good matching along the horizontal and vertical axes (represented as blue and red lines). However, the *spring* and *sinusoidal* models are of one-dimensional nature (e.g., force F_x only depends on x distance ($u_x - p_x$)) and become inaccurate at points deviating from the main axes.

4.1.2 Our Model. The complexity of the force distribution modelled by the Gor'kov potential increases as the distance to the centre of the trap increases. However, we only need to derive a model allowing us to predict where to place the trap to produce a specific force on the particle. This allows us to limit our considerations to the region corresponding to the peaks designated by $\mathcal{A}_r$ and $\mathcal{A}_z$ in Figure 5.

For points within this region, forces around twin traps distribute in a mostly *axis-symmetric* fashion, which can be approximated as

$$F_r(\mathbf{p}, \mathbf{u}) := \mathcal{A}_r \cdot \cos(\mathcal{V}_z \cdot (u_z - p_z)) \quad (4a)$$

$$\cdot \sin\left(\mathcal{V}_{xr} \cdot \sqrt{(u_x - p_x)^2 + (u_y - p_y)^2}\right),$$

$$F_z(\mathbf{p}, \mathbf{u}) := \mathcal{A}_z \cdot \sin(\mathcal{V}_z \cdot (u_z - p_z)) \quad (4b)$$

$$\cdot \cos\left(\mathcal{V}_{zr} \cdot \sqrt{(u_x - p_x)^2 + (u_y - p_y)^2}\right),$$

where $\mathcal{A}_r, \mathcal{A}_z$ represent peak trapping forces along the radial and vertical directions of the trap, respectively. $\mathcal{V}_z, \mathcal{V}_{xr}, \mathcal{V}_{zr}$ represent characteristic frequencies of the sinusoids describing how the forces evolve around the trap. The resulting forces can be converted into acoustic forces in 3D space from these cylindrical coordinates, with azimuth $\phi = \arctan((u_y - p_y)/(u_x - p_x))$:

$$F(\mathbf{p}, \mathbf{u}) = \begin{pmatrix} F_x(\mathbf{p}, \mathbf{u}) \\ F_y(\mathbf{p}, \mathbf{u}) \\ F_z(\mathbf{p}, \mathbf{u}) \end{pmatrix} := \begin{pmatrix} F_r(\mathbf{p}, \mathbf{u}) \cos \phi \\ F_r(\mathbf{p}, \mathbf{u}) \sin \phi \\ F_z(\mathbf{p}, \mathbf{u}) \end{pmatrix}. \quad (5)$$

We validated our model by comparing its accuracy against the forces predicted by the gradient of the Gor'kov potential. More specifically, we simulated 729 single traps homogeneously distributed across the working volume of our levitator (i.e., $8 \times 8 \times 8$ cm, in line with [Fushimi et al. 2019; Hirayama et al. 2019]), testing 400 points around each trap for a total of 400×729 force estimations.

Table 1 summarises the error distribution achieved by these three models when compared to Gor'kov, showing an average relative error as low as 4% for our model and much poorer fitting for the other two models. Full details and raw data used in this validation can be found in the Supplementary Material S2.

As a final summary, this results in a model for our trap-particle dynamics that only depends on $\mathbf{p}$ and $\mathbf{u}$ and which fits the

Table 1. Fit Parameters for the Spring, Sinusoidal and Axis-symmetric Models and the Respective Average Relative Errors When Compared to Gor'kov

Model	Spring		Sinusoidal		Axis-symmetric	
Fit	$\mathcal{K}_x$	-0.0071	$\mathcal{A}_x$	0.00009	$\mathcal{A}_r$	0.0004636
	$\mathcal{K}_y$	-0.0071	$\mathcal{A}_y$	0.00009	$\mathcal{A}_z$	0.0002758
	$\mathcal{K}_z$	-0.94	$\mathcal{A}_z$	-0.0019	$\mathcal{V}_z$	1307.83
			$\mathcal{V}_x$	-68.92	$\mathcal{V}_{xr}$	-476.49
			$\mathcal{V}_y$	-68.92	$\mathcal{V}_{zr}$	287.87
			$\mathcal{V}_z$	1307.83		
Error	63.3%		35.9%		4.3%	

definition of a second-order ordinary differential equation

$$m\ddot{\mathbf{p}}(t) = F(\mathbf{p}(t), \mathbf{u}(t)). \quad (6)$$

While the resulting model is accurate (4% relative error), we note that it is not invertible, which will complicate the formulation of our solution approach. For instance, for a particle at $\mathbf{p} = (0, 0, 0)$, force $F(\mathbf{p}, \mathbf{u}) = (0, 0, \mathcal{A}_z \cdot \cos(\mathcal{V}_{zr} \cdot x))$ can be obtained with either $\mathbf{u} = (x, 0, \pi/(2\mathcal{V}_z))$ or $\mathbf{u} = (-x, 0, \pi/(2\mathcal{V}_z))$.

4.2 Open-Loop Optimal Path Following

This section computes the timing for the particle, so that it moves along the given reference path Q from Equation (1) in minimum time and according to the dynamics of the system. We approach this as an open-loop (or feed-forward) path following problem, as typically done in robotics [Faulwasser 2012]. That is, we design an OCP that computes the timing $t \mapsto \theta(t)$ as to keep the levitated particle on the prescribed path, to traverse the path in optimum (minimum) time while considering the trap-particle dynamics (i.e., minimum *feasible* time).

Our non-invertible model of particle dynamics calls for a more complex treatment of the problem, which we split into two parts. In the first part, we derive a virtual system for the timing law, which we describe as a system of first-order differential equations, solvable with traditional methods. In the second part, we couple the timing law with our non-invertible dynamics, introducing auxiliary variables that will later enable pseudo-inversion (i.e., to compute trap placement for a given force) as described in Section 4.3.

4.2.1 Error Dynamics and the Timing Law. The requirement that the particle follows the path Q exactly and for all times means that the deviation from the path equals zero for all $t \in \mathbb{R}_0^+$, i.e.,

$$\mathbf{e}(t) := \mathbf{p}(t) - \mathbf{q}(\theta(t)) \equiv 0.$$

If the path deviation $\mathbf{e}(t)$ is 0 during the whole interval $[t_0, t_1]$, this implies that the time derivatives of $\mathbf{e}(t)$ also have to vanish on (t_0, t_1) . Considering this (i.e., $\dot{\mathbf{e}}(t) \equiv 0$ and $\ddot{\mathbf{e}}(t) \equiv 0$) results in

$$\dot{\mathbf{p}}(t) = \dot{\mathbf{q}}(\theta(t)), \quad (7a)$$

$$\ddot{\mathbf{p}}(t) = \ddot{\mathbf{q}}(\theta(t)) = \frac{\partial \dot{\mathbf{q}}}{\partial \theta} \dot{\theta}(t), \quad (7b)$$

$$\ddot{\mathbf{p}}(t) = \ddot{\mathbf{q}}(\theta(t)) = \frac{\partial^2 \mathbf{q}}{\partial \theta^2} \dot{\theta}(t)^2 + \frac{\partial \mathbf{q}}{\partial \theta} \ddot{\theta}(t). \quad (7c)$$

Thus, for particles on the path Q , system dynamics of the form (6), and provided we are able to express $\mathbf{u}$ as a function of

$\mathbf{p}$ and $\ddot{\mathbf{p}}$ (i.e., the system inversion described in Section 3.2), the position, the velocity, and the acceleration of the particle can be expressed via θ , $\dot{\theta}$, $\ddot{\theta}$, respectively. For details, a formal derivation, and tutorial introductions we refer to [Faulwasser 2012; Faulwasser et al. 2017].

Observe that in Equation (7) the partial derivatives $\frac{\partial^2 \mathbf{q}}{\partial \theta^2}$ and $\frac{\partial \mathbf{q}}{\partial \theta}$, as well as $\dot{\theta}$ and $\ddot{\theta}$ appear. This leads to two additional observations: First, the parametrisation $\mathbf{q}$ from Equation (1) should be at least twice continuously differentiable with respect to θ , so that the partial derivatives are well-defined. This justifies our constraint in Section 2. Please note this does not rule out corners in the path Q , as shown by the parametrisation Equation (2) of the cardioid. Second, the time evolution of θ should be continuously differentiable, as otherwise large jumps can occur in the acceleration.

To avoid said jumps, we generate the timing $t \mapsto \theta(t)$ via the double integrator

$$\ddot{\theta}(t) = v(t), \quad (8)$$

where $v(t) \in \mathbb{R}$ is a computational degree of freedom, used to control the progress of the particle along Q . The function $v(t)$ enables us to later cast the computation of time-optimal motions along reference paths as an OCP, see Equation (16).

The in-homogeneous second-order ordinary differential Equation (8) takes care of jumps, but needs to be augmented by conditions on θ and $\dot{\theta}$ at initial time $t = 0$ and final time $t = T$. This is done to represent the periodic nature of our content (i.e., the particle reveals the same path many times per second):

$$\theta(0) = \theta_0, \quad \theta(T) = \theta_f, \quad \dot{\theta}(0) = \dot{\theta}(T), \quad (9)$$

where θ_0 and θ_f are taken from Equation (1) and the total time T will be optimally determined by the OCP. The first two equations in (9) ensure that the path Q is fully traversed, while the third one ensures the speeds at the beginning and end of the path match.

With these additional considerations, we define the (state) vector $\mathbf{z}(t) := (\theta(t), \dot{\theta}(t))^T$, which finally allows us rewrite our second-order differential equations in (8) as a system of first-order differential equations

$$\dot{\mathbf{z}}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{z}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v(t), \quad \mathbf{z}(0) = \mathbf{z}_0, \quad \mathbf{z}(T) = \mathbf{z}_T. \quad (10)$$

Please note that (10) is an equivalent *virtual* system to (8) and (9), solvable using standard Runge-Kutta methods [Butcher 2016]. The system (10) is suitable to generate the timing law, but the particle dynamics (6) still need to be included, as described next.

4.2.2 Coupling of Non-invertible Particle Dynamics. To include the particle dynamics (6) in the computation of the timing, we need to couple them with the system (10). The typical approach is to rewrite Equation (6) as

$$M(\ddot{\mathbf{p}}(t), \mathbf{p}(t), \mathbf{u}(t)) := m\ddot{\mathbf{p}}(t) - F(\mathbf{p}(t), \mathbf{u}(t)) = 0, \quad (11)$$

and make sure that this equation locally admits an inverse function:

$$\mathbf{u}(t) = M^{-1}(\ddot{\mathbf{p}}(t), \mathbf{p}(t)). \quad (12)$$

This would allow us to easily compute the location of our traps. However, such inversion is not straightforward for our model, and

to the best of our knowledge, standard solution methods do not exist for such cases.

We deal with this challenge by introducing constraints related to our particle dynamics. These will allow us to compute feasible timings while delaying the computation of the exact trap location to a later stage in the process.

More specifically, we introduce auxiliary variables $\zeta_1, \dots, \zeta_6$ for each trigonometric term in the force F , and we replace $p_i = q_i(\theta)$ for $i \in \{x, y, z\}$ (i.e., particle position exactly matches our reference path)

$$\zeta_1 = \sin\left(\mathcal{V}_{xr} \sqrt{(u_x - q_x(\theta))^2 + (u_y - q_y(\theta))^2}\right), \quad (13a)$$

$$\zeta_2 = \cos(\mathcal{V}_z \cdot (u_z - q_z(\theta))), \quad (13b)$$

$$\zeta_3 = \sin(\mathcal{V}_z \cdot (u_z - q_z(\theta))), \quad (13c)$$

$$\zeta_4 = \cos\left(\mathcal{V}_{xr} \sqrt{(u_x - q_x(\theta))^2 + (u_y - q_y(\theta))^2}\right), \quad (13d)$$

$$\zeta_5 = \sin \phi, \quad (13e)$$

$$\zeta_6 = \cos \phi. \quad (13f)$$

With this, we are now able to formally express the force (5) in terms of $\zeta := (\zeta_1, \dots, \zeta_6)$, i.e.,

$$\tilde{F}(\zeta) := \begin{pmatrix} \mathcal{A}_r \zeta_1 \zeta_2 \zeta_6 \\ \mathcal{A}_r \zeta_1 \zeta_2 \zeta_5 \\ \mathcal{A}_z \zeta_4 \zeta_3 \end{pmatrix}. \quad (14)$$

Using these auxiliary variables, we can now couple the trap-particle dynamics (6) with the virtual system (10). To this end, similar to (11), we define the following constraint along path Q :

$$\tilde{M}(\theta(t), \dot{\theta}(t), v(t), \zeta(t)) := m\ddot{\mathbf{q}}(\theta(t)) - \tilde{F}(\zeta(t)) = 0. \quad (15)$$

Observe that (i) we need $\dot{\theta}(t)$ due to (7c); and (ii) $\ddot{\theta}(t)$ can be replaced by $v(t)$ due to Equation (8).

Finally, combining this with the prior first-order differential system allows us to conceptually formulate the problem of computing a minimum-time motion along the path Q as

$$\begin{aligned} & \min_{v, T, \zeta} T + \gamma \int_0^T v(t)^2 dt \\ & \text{subject to} \\ & \dot{\mathbf{z}}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{z}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v(t), \quad \mathbf{z}(0) = \mathbf{z}_0, \quad \mathbf{z}(T) = \mathbf{z}_T, \\ & 0 = \tilde{M}(\mathbf{z}(t), v(t), \zeta(t)), \\ & \zeta(t) \in [-1, 1]^6. \end{aligned} \quad (16)$$

The constraint on $\zeta(t)$ is added since the trigonometric structure of (13) is not directly encoded in the OCP, while $\gamma \geq 0$ is a regularisation parameter.

The case $\gamma = 0$ corresponds to strictly minimising time, which typically leads to an aggressive use of the forces around the trap. The example in Figure 6(a) shows the trap accelerating the particle before arriving at the corner, then applying aggressive deceleration before it reaches the corner. At the top and bottom of the shape, again the particle is accelerated strongly, then it is suddenly decelerated by the traps placed behind it, in order to move around the curve. This would be the optimum solution in an ideal case (i.e., a device working *exactly* as per our model), but inaccuracies in the

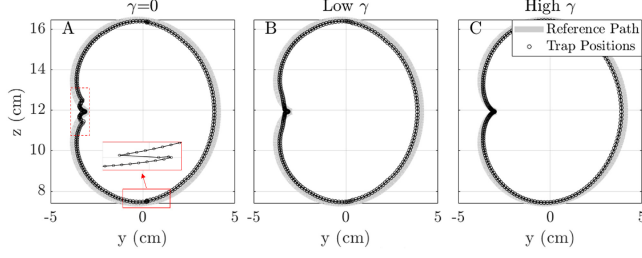


Fig. 6. Effect of the regularisation parameter γ on the position of the optimised traps.

real device can make them unstable. Moreover, we observed the overall reductions in rendering time to usually be quite small.

The regularisation ($\gamma > 0$) enables *nearly* time-optimal solutions. More specifically, this is a strictly convex regularisation that penalises high magnitudes of the virtual input $v = \ddot{\theta}$. This is most closely related to the accelerations applied to the particle, hence avoiding aggressive acceleration/deceleration as shown in Figure 6(b) and (c). Several heuristics could be proposed to automate the selection of a suitable value for γ (e.g., use smallest γ ensuring that the dot product of $\dot{\mathbf{p}}(t)$ and $\dot{\mathbf{u}}(t)$ remains always positive), but this step is considered beyond the scope of the current article.

Note that the OCP in Equation (16) yields the (nearly) optimal timing along the path Q , respecting the particle dynamics and hence solving *Challenge 1*. Moreover, it yields the required forces through $\zeta(t)$. However, it does not give the trap trajectory $\mathbf{u}(t)$ that generates these forces. To obtain the trap trajectory and thus solve *Challenge 2*, we refine Equation (16) in the following section.

4.3 Computing the Trap Trajectory

The second stage in our approach deals with *Challenge 2*, computing the trap trajectory $\mathbf{u}(t)$ based on the solution of Equation (16). In theory, we would use the values for $\zeta_i(t)$, $i = 1, \dots, 6$ and $q_j(\theta(t))$, $j \in \{x, y, z\}$ to determine the particle position and force required at each moment in time, obtaining the required trap position $\mathbf{u}(t)$ by solving Equation (13).

In practice, solving $\mathbf{u}(t)$ from Equation (13) numerically is not trivial, particularly for values of ζ_i close to ± 1 , where numerical instabilities could occur, particularly for ζ_1 and ζ_4 .

We attenuate these difficulties by providing further structure to the OCP (16), specifically regarding ζ :

$$\zeta_2^2 + \zeta_3^2 = 1, \quad \zeta_5^2 + \zeta_6^2 = 1. \quad (17)$$

We also constrain the solvability of Equation (13) by using a constant back-off $\varepsilon \in]0, 1[$ in order to avoid numerical instabilities:

$$-1 + \varepsilon \leq \zeta_i \leq 1 - \varepsilon, \quad i = 1, \dots, 6. \quad (18)$$

For the sake of compact notation, we summarise the above constraints in the following set notation

$$\mathcal{Z} := \left\{ \zeta \in \mathbb{R}^6 \mid (17) \text{ and } (18) \text{ are satisfied} \right\}. \quad (19)$$

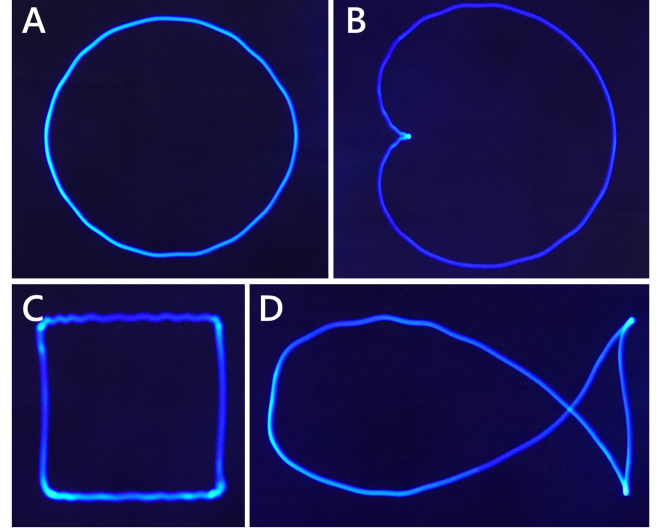


Fig. 7. Test shapes used for evaluation, as rendered by our *OptiTrap* approach. The circle (A) provides a trivial timing case. The cardioid (B) and squire (C) represent simple shapes featuring sharp corners and straight lines. The fish (D) is used as an example of a composite shape featuring corners, straight lines, and curves.

The final OCP is then given by

$$\min_{v, T, \zeta} T + \gamma \int_0^T v(t)^2 dt$$

subject to

$$\begin{aligned} \dot{\mathbf{z}}(t) &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{z}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v(t), \quad \mathbf{z}(0) = \mathbf{z}_0, \quad \mathbf{z}(T) = \mathbf{z}_T, \\ 0 &= \tilde{M}(\mathbf{z}(t), v(t), \zeta(t)), \\ \zeta(t) &\in \mathcal{Z}. \end{aligned} \quad (20)$$

Finally, given $\zeta(t) \in \mathcal{Z}$ and $\theta(t)$ from the solution of Equation (20), we numerically solve Equation (13) for u_i , $i \in \{x, y, z\}$, using Powell's dog leg method [Mangasarian 1994]. Please note that higher values of ε will limit the magnitudes of the forces exploited by our approach. A simple solution is to perform a linear search over ε , only increasing its value and recomputing the solution if Powell's method fails to converge to a feasible trap location. For details on discretising the OCP (20), we refer to the Supplementary Material S3.

5 EVALUATION

This section evaluates the capability of our algorithm to support content creation by generating physically feasible trap trajectories given only a reference path (i.e., a shape) as an input. We select a range of shapes and analyse the effects each step in our approach has on the final achievable size, rendering frequency, and reconstruction error, for each shape. We also inspect how these steps influence the presence of visual distortions in the end result. Finally, we analyse the resulting acceleration profiles.

5.1 Test Shapes

We used four test shapes for the evaluation: a circle, a cardioid, a squircle, and a fish, all shown in Figure 7.

The circle is selected as a trivial case in terms of timing. It can be optimally sampled by using a constant angular speed (i.e., constant acceleration), and is there to test whether our solution converges towards such optimum solutions. The cardioid and squircle represent simple shapes featuring sharp corners (cardioid) and straight lines (squircle), both of them challenging features. Finally, the fish is selected as an example composite shape, featuring all elements (i.e., corners, straight lines, and curves).

All of these shapes can be parameterised by low-frequency sinusoids (see Supplementary Material S4), ensuring the path parameter θ progresses slowly in areas of high curvature. This provides a good starting point for our baseline comparison that we explain in Section 5.2. While our approach works for content in 3D, we perform the evaluation on shapes in a 2D plane of the 3D space to facilitate the analysis of accelerations in Section 5.6, which we will perform in terms of horizontal and vertical accelerations (i.e., the independent factors given our axis-symmetric model of forces), allowing us to validate *OptiTrap*'s awareness of particle directions and how close it can get to the maximum accelerations allowed by the dynamics of acoustic traps, as discussed in Section 4.1.

Notice in Figure 7, that due to the timing differences, different shape elements exhibit different levels of brightness. To obtain homogeneous brightness along the shape, please refer to the solution proposed by [Hirayama et al. 2019], where the particle illumination is adjusted to the particle speed.

5.2 Conditions Compared

We compare three approaches to rendering levitated shapes, where each approach subsequently addresses one of the challenges introduced in Section 2. This will help us assess the impact that each challenge has on the final results obtained.

The first condition is a straight-forward *Baseline*, with homogeneous sampling of the path parameter, which matches the example strategy shown in Figure 3(b), and placing traps where the particle should be. The *Baseline* still does not address any of the challenges identified (i.e., optimum timing or trap placement), but it matches approaches used in previous works [Fushimi et al. 2020; Hirayama et al. 2019; Plasencia et al. 2020] and will illustrate their dependence on the specific shape and initial parametrisation used.

The second condition, *OCP_Timing*, makes use of our *OptiTrap* approach to compute optimum and feasible timing (see Section 4.2.1), but still ignores *Challenge 2*, assuming that particles match trap location (i.e., it skips Section 4.2.2). The third condition is the full *OptiTrap* approach, which considers feasibility and trap-particle dynamics and deals with both the timing and trap placement challenges.

These conditions are used in a range of comparisons involving size, frequency, reconstruction error as well as comparative analysis of the effects of each condition on the resulting particle motion, detailed in the following sections.

5.3 Maximum Achievable Sizes

This section focuses on the maximum sizes that can be achieved for each test shape and condition while retaining an overall rendering

time of 100 ms¹ (i.e., PoV threshold). For each of these cases, we report the maximum achievable size and reliability, which were determined as follows.

Maximum size is reported in terms of shape width and in terms of meters of *content per second* rendered [Plasencia et al. 2020], and their determination was connected to the feasibility of the trap trajectories, particularly for the *Baseline* condition.

More specifically, we determined maximum feasible sizes for the *Baseline* condition by conducting an iterative search. That is, we increased the size at each step and tested the reliability of the resulting shape. We considered the shape feasible if it could be successfully rendered at least 9 out of 10 times in the actual levitator. On success, we would increase the size by increasing the shape width by half a centimetre. On a failure, we would perform a binary search between the smallest size failure and the largest size success, stopping after two consecutive failures, and reporting the largest successful size. This illustrates the kind of trial and error that a content designer would need to go through without our approach and it was the most time-consuming part of this evaluation.

For the other two conditions (i.e., *OCP_Timing* and *OptiTrap*), maximum sizes could be determined without needing to validate their feasibility in the final device. Again, we iteratively increased the shape size using the same search criteria as before (i.e., 5 mm increases, binary search). We assumed any result provided would be feasible (which is a reasonable assumption; see below) and checked the resulting total rendering time, increasing the size if the time was still less than 100 ms. At each step, a linear search was used, iteratively increasing ϵ , until feasible trap locations could be found (i.e., Equation (13) could be solved without numerical instabilities). Adjusting the value of the regularisation parameter γ was done by visually inspecting the resulting trap trajectories, until no discontinuities could be observed (see Figure 6). Please note that this is a simple and quick task, which, unlike the *Baseline* condition, does not involve actual testing on the levitation device and could even be automated. All solutions provided can be assumed feasible, and the designer only needs to choose the one that better fits their needs.

Once the maximum achievable sizes were determined, these were tested in the actual device. We determined the feasibility of each shape and condition by conducting ten tests and reporting the number of cases where the particle succeeded to reveal the shape. This included the particle accelerating from rest, traversing/revealing the shape for 6 seconds, and returning to rest.

Table 2 summarises the results achieved for each test shape and condition. First of all, it is worth noting that all trials were successful for *OptiTrap* and *OCP_Timing* approaches, confirming our assumption that the resulting paths are indeed feasible and underpinning *OptiTrap*'s ability to avoid trial and error on the actual device during content creation.

The results and relative improvements in terms of size vary according to the particular condition and shape considered. For instance, it is interesting to see that all conditions yield similar final sizes for the circle, showing that *OptiTrap* (and *OCP_Timing*) indeed converge towards optimum solutions.

¹The circle rendered at 100 ms would exceed the size of our device's working volume. Thus, this shape was rendered at 67 ms.

Table 2. Maximum Shape Width and Meters of Content Per Second Achieved with the *Baseline*, *OCP_Timing*, and *OptiTrap* Approach at Constant Rendering Frequency

Shape	Freq. (Hz)	<i>Baseline</i>			<i>OCP_Timing</i>			<i>OptiTrap</i>			Increase in Width w.r.t. the <i>Baseline</i>
		Width (cm)	Content per Second (m)	Success Rate	Width (cm)	Content per Second (m)	Success Rate	Width (cm)	Content per Second (m)	Success Rate	
Circle	15	7.00	3.30	10/10	7.00	3.30	10/10	7.00	3.30	10/10	0%
Cardioid	10	8.05	2.48	10/10	9.09	2.80	10/10	9.09	2.80	10/10	12.9%
Squircle	10	0.80	0.29	9/10	5.30	1.90	10/10	5.30	1.90	10/10	562.5%
Fish	10	7.77	2.42	10/10	8.76	2.75	10/10	8.76	2.75	10/10	12.7%

Successful trials out of 10.

Table 3. Maximum Rendering Frequencies Achieved with the *Baseline*, *OCP_Timing*, and *OptiTrap* Approach, for Shapes of Equal Size

Shape	<i>Baseline</i>		<i>OCP_Timing</i>		<i>OptiTrap</i>	
	Freq. (Hz)	Success Rate	Freq. (Hz)	Success Rate	Freq. (Hz)	Success Rate
Circle	15	10/10	15	10/10	15	10/10
Cardioid	8	10/10	10	10/10	10	10/10
Squircle	4	10/10	10	10/10	10	10/10
Fish	9	9/10	10	10/10	10	10/10

Successful trials out of 10.

More complex shapes where the *Baseline* parametrisation is not optimal (i.e., cardioid, squircle, and fish), show increases in size when using *OCP_Timing* and *OptiTrap*. More interestingly, the increases in size vary greatly between shapes, showing increases of around 12% for the fish and cardioid, and up to 562% for the squircle. This is the result of the explicit parametrisation used, with reduced speeds at corners in the fish and cardioid cases, but not in the case of the squircle.

It is also interesting to see that *OCP_Timing* and *OptiTrap* maintain high values of *content per second* rendered, independently of the shape. That is, while the performance of the *Baseline* approach is heavily determined by the specific shape (and parametrisation) used, the OCP-based approaches (*OCP_Timing* and *OptiTrap*) yield results with consistent *content per second*, determined by the capabilities of the device (but not so much by the specific shape). Finally, please note that no changes in terms of maximum size can be observed between *OCP_Timing* and *OptiTrap*.

5.4 Maximum Rendering Frequencies

In this second evaluation, we assessed the effect of the timing strategy on the maximum achievable rendering frequencies. To do this, we selected the maximum achievable sizes obtained in the prior evaluation for each shape. We then reproduced such sizes with the *Baseline* approach, searching for the maximum frequency at which this approach could reliably render the shape (using the same searching and acceptance criteria as above).

That is, while we knew *OCP_Timing* and *OptiTrap* could provide 10 Hz for these shapes and sizes, we wanted to determine the maximum frequency at which the *Baseline* would render them, as to characterise the benefits provided by optimising the timing.

The results are summarised in Table 3. As expected, no changes are produced for the circle. However, there is a 25% increase for the

Table 4. RMSE and **Path-normalised (PN)** RMSE with Respect to the Total Path Length of Each Test Shape, for the *Baseline*, *OCP_Timing*, and *OptiTrap* Approach, at Constant Rendering Frequency

Shape	<i>Baseline</i>		<i>OCP_Timing</i>		<i>OptiTrap</i>	
	RMSE (cm)	PN RMSE	RMSE (cm)	PN RMSE	RMSE (cm)	PN RMSE
Circle	0.196	0.893	0.146	0.666	0.114	0.521
Cardioid	0.142	0.575	0.153	0.545	0.120	0.429
Squircle	0.056	1.956	0.0824	0.434	0.080	0.421
Fish	0.111	0.458	0.147	0.536	0.0581	0.212

cardioid, 150% for the squircle, and 11% increase for the fish using the *OptiTrap* method. Again, please note that no differences can be observed in terms of maximum frequency between *OCP_Timing* and *OptiTrap*.

5.5 Reconstruction Accuracy

To evaluate the reconstruction accuracy, we recorded each of the trials in our maximum size evaluations (see Section 5.3) using an OptiTrack camera system. We then computed the **Root Mean Squared Error (RMSE)** between the recorded data and the intended target shape, taking 2s of shape rendering into account (exclusively during the cyclic part, ignoring ramp-up/ramp-down). For a fair comparison across trials, we normalise the RMSE with respect to the traversed paths, by dividing the RMSE by the total path length of each individual test shape. The results are summarised in Table 4.

On average, both *OCP_Timing* and *OptiTrap* provide better results in terms of accuracy, when compared to the *Baseline*. It is particularly worth noting that the accuracy results, in terms of the raw RMSE for *OptiTrap* and *OCP_Timing*, are better for the cardioid when compared to the *Baseline*, even though a larger shape is being rendered. While the raw RMSE of *OCP_Timing* and *OptiTrap* is slightly larger for the squircle, we need to take into account that the size rendered by *OCP_Timing* and *OptiTrap* is almost 6 times larger. Comparing accuracy in terms of normalised RMSE shows consistent increases of accuracy for *OptiTrap* compared to the *Baseline*, with overall decreases in the RMSE of 41.7% (circle), 25.4% (cardioid), 78.5% (squircle), and 53.8% (fish). *OCP_Timing* shows a decrease of 25.4%, 5.14%, and 77.8% of the normalised RMSE for the circle, cardioid, and squircle, with the exception of the fish, where the normalised RMSE increased by 17.1%, when compared to the *Baseline*.

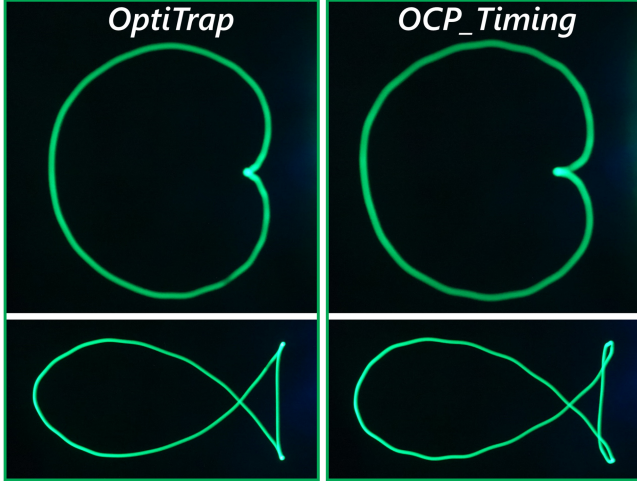


Fig. 8. Visual comparison between shapes rendered with traps located according to particle dynamics using *OptiTrap* (left) and traps placed along the reference path using *OCP_Timing* (right) for the cardioid (top) and the fish (bottom) test shapes. Please note undesired increases in size and error in sharp features, such as corners.

Comparing the reconstruction accuracy of *OCP_Timing* and *OptiTrap* highlights the relevance of considering trap dynamics to determine trap locations (i.e., Section 4.3). *OptiTrap* consistently provides smaller RMSE than *OCP_Timing*, as a result of considering and accounting for the trap-to-particle displacements required to apply specific accelerations. We obtain a 21.8%, 21.3%, 2.9%, and 60.5% decrease in the normalised RMSE, for the circle, cardioid, squircle, and fish, respectively. Such differences are visually illustrated in Figure 8, showing the effects on the cardioid and fish, for the *OCP_Timing* and *OptiTrap* approaches. It is worth noting how placing the traps along the reference path results in the cardioid being horizontally stretched, as the particle needs to retain larger distances to the trap to keep the required acceleration (horizontal forces are weaker than vertical ones). This also results in overshooting of the particle at corner locations, which can be easily observed at the corner of the cardioid and in the fins of the fish. For completeness, the visual comparison for the remaining two test shapes is provided in the Supplementary Material S5.

5.6 Analysis of Acceleration Profiles

Finally, we examined the acceleration profiles produced by *OptiTrap* and how these differ from the pre-determined acceleration profiles of the *Baseline*. Please note that we only compare and discuss *OptiTrap* and *Baseline* in Figure 9, and only for the fish shape. *OCP_Timing* is not included, as it results in similar acceleration profiles as *OptiTrap*. Only the fish is discussed, as it already allows us to describe the key observations that can be derived from our analysis. For completeness, the acceleration profiles for all remaining test shapes are included in the Supplementary Material S6.

As introduced above, Figure 9 shows the particle accelerations and speeds of a particle revealing a fish shape of maximum size (i.e., a width of 8.76 cm), rendered over 100 ms using the *OptiTrap*

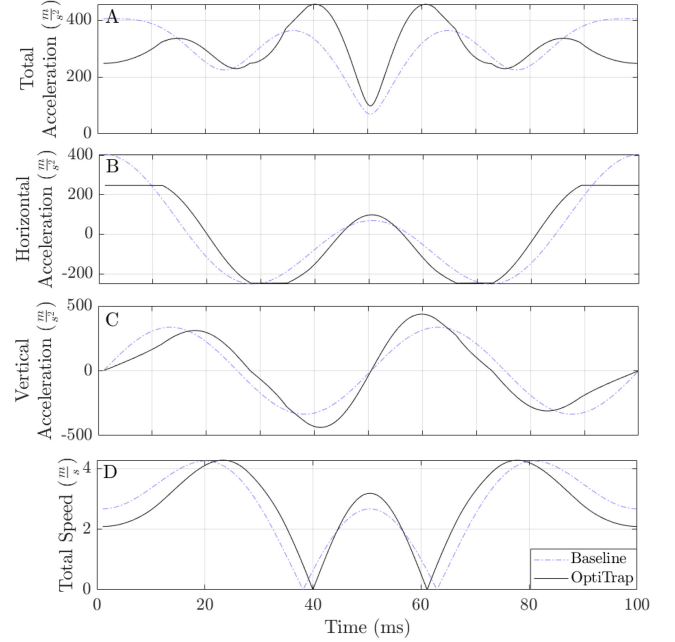


Fig. 9. Particle acceleration and speed for the fish rendered at 10 Hz using *OptiTrap* (solid black) and the *Baseline* (dash-dotted purple). While the speed profiles (D) of both trap trajectories are similar, the trap trajectory generated by our approach is feasible, whereas the one generated by the *Baseline* is not. The main reason is that the *Baseline* exceeds the feasible horizontal acceleration, while our approach caps it to feasible values (B). Our approach compensates by using higher (available) vertical acceleration (C). Note that the times where the *Baseline* applies a higher total acceleration (A) than our approach are those where our approach respects the constraints on the feasible horizontal acceleration (B).

and *Baseline* approaches. It is worth noting that while *OptiTrap* succeeded in rendering this shape, *Baseline* did not (the maximum width for *Baseline* was 7.77 cm).

A first interesting observation is that although *OptiTrap* provides lower total accelerations than the *Baseline* during some parts of the path (see Figure 9(a)), it still manages to reveal the shape in the same time. The key observation here is that the regions where the total acceleration is lower for *OptiTrap* match with the parts of the path where the horizontal acceleration is very close to its maximum; for example, note the flat regions in Figure 9(b) of around $\pm 300 \text{ m/s}^2$. This is an example of *OptiTrap*'s awareness of the dynamics and capabilities of the actual device, limiting the acceleration applied to retain the feasibility of the path.

Second, it is worth noting that maximum horizontal accelerations are significantly smaller than vertical accelerations. As such, it does make sense for horizontal displacements to become the limiting factor. In any case, neither acceleration exceeds its respective maximum value. The *OptiTrap* approach recovers any missing time by better exploiting areas where acceleration is unnecessarily small. It is also worth noting that the value of the horizontal and vertical accelerations are not simply being capped to a maximum acceleration value per direction (e.g., simultaneously maxing out at $\pm 300 \text{ m/s}^2$ and $\pm 600 \text{ m/s}^2$ in the horizontal and vertical directions),

as such cases are not feasible according to the dynamics of our acoustic traps.

Third, it is interesting to note that the final acceleration profile is complex, retaining little resemblance with the initial parametrisation used. This is not exclusive for this shape and can be observed even in relatively simple shapes, such as the cardioid in Figure 3(d) or the remaining shapes, provided in the Supplementary Material S6.

The complexity of these profiles is even more striking if we look at the final speeds resulting from both approaches (see Figure 9(d)). Even if the acceleration profiles of *Baseline* and *OptiTrap* are very different, their velocity profiles show only relatively subtle differences. But even if these differences are subtle, they mark the difference between a feasible path (i.e., *OptiTrap*) and a failed one (i.e., *Baseline*). These complex yet subtle differences also illustrate how it is simply not sensible to expect content designers to deal with such complexity, and how *OptiTrap* is a necessary tool to enable effective exploitation of PoV levitated content.

6 DISCUSSION

This article presented *OptiTrap*, an automated approach for optimising timings and trap placements, to achieve feasible target shapes. We believe this is a particularly relevant step for the adoption of levitation PoV displays, as it allows the content creator to focus on the shapes to present, with feasible solutions being computed automatically while making effective usage of the capabilities of the device. As such, we hope *OptiTrap* to become an instrumental tool in helping explore the actual potential of these displays.

However, *OptiTrap* is far from a complete content creation tool. Such a tool should consider the artist's workflows and practices. Similarly, testing and identifying the most useful heuristics to tune our approach (e.g., the regularisation) or visualisation tools identifying tricky parts of the shapes (i.e., requiring high accelerations) should be included as a part of this process. Our goal is simply to provide the base approach enabling this kind of tool.

Even this base approach can be extended in a variety of ways. Our levitator prototype is built from off-the-shelf hardware, and is still subject to inaccuracies that result in distortions in the sound-fields generated [Fushimi et al. 2019]. As such, more accurate levitation hardware or, alternatively, a model reflecting the dynamics of the system in a more accurate manner would be the most obvious pathway to improve our approach. It is worth noting that both factors should be advanced jointly. A more accurate model could also be less numerically stable, potentially leading to worse results if the hardware is not accurate enough.

The high update rates of 10 kHz required by the levitator and the millisecond delays introduced by optical tracking systems indicate that closed-loop approaches can be both promising and challenging avenues to explore. Assuming a tracking device synchronised with the levitation device (as to map current trap locations with real particle positions), *OptiTrap* could be combined with learning-based approaches. These learning-based approaches will require example paths (i.e., with initial timing and trap placement), and their achievable complexity and convergence will be limited by the examples provided. In such cases, *OptiTrap* can be used to always generate feasible initial trajectories (i.e., to avoid system restarts on

failure). Thus learning-based approaches could be used to further refine *OptiTrap* beyond our current model, as to account for device inaccuracies such as those discussed by [Fushimi et al. 2019].

However, higher gains can be obtained from more radical changes in the approach. For instance, our approach optimises the timing and trap placement, but it does not modify the target shapes. As illustrated in Figure 9, slight changes in speed have significant effects on the acceleration profiles (and feasibility) of the shapes. This is even more prominent for the position, where small changes can heavily influence the acceleration and feasibility of target shapes. As such, approaches exploiting subtle modifications to the shape could lead to significant gains in rendering performance.

Another interesting possibility would be extending our approach to use several particles. As shown in [Plasencia et al. 2020], while the use of several particles does not increase the overall power that can be leveraged, it does allow for increased flexibility. That is, the intensity/forces of each trap can be individually and dynamically adjusted, to match the needs of the region of the path that each particle is revealing. Also, particles can each be rendering specific independent features, so the content is not limited to a single connected path, and the particles do not waste time/accelerations traversing parts of the path that will not be illuminated (i.e., visible). This approach, however, entails significant challenges. The first obvious challenge is the reliability of the intensity control of the traps. [Plasencia et al. 2020] demonstrate accurate control of the stiffness at the centre of the traps, but the effects of multiple (interfering) traps in each trap's topology (i.e., how forces distribute around the trap) is yet to be studied. A second challenge is that each particle is not forced to traverse the path at the same speed/rates, with such independent timing progression becoming an additional degree of freedom to account for.

Finally, further extensions to our work can come from its application to domains other than PoV displays. An obvious next step would be to adapt *OptiTrap* to photophoretic displays, which trap particles using optical traps instead of acoustic traps [Kumagai et al. 2021; Smalley et al. 2018]. This would involve including a model of the dynamics of such optical traps, but it can also involve further challenges, such as modelling the response times of galvanometers and LC panels involved in creating the trap.

Our approach can also be adapted to applications requiring objects to be transported quickly and accurately. For example, contactless transportation of matter has a wealth of applications in areas such as the study of physical phenomena, biochemical processes, materials processing, or pharmaceuticals [Foresti et al. 2013]. Our method can help solve such problems by directly computing a rest-to-rest solution of the matter to be transported, given an estimation of the mass of that matter, the acoustic force acting on the matter, and a path from start to target position.

7 CONCLUSION

In this article, we proposed the first structured numerical approach to compute trap trajectories for acoustic levitation displays. *OptiTrap* automatically computes physically feasible and nearly time-optimal trap trajectories to reveal generic mid-air shapes, given

only a reference path. Building on a novel multi-dimensional approximation of the acoustic forces around the trap, we formulate and show how to solve a non-linear path following problem without requiring or exploiting differential flatness of the system dynamics. We demonstrate increases of up to 563% in size and up to 150% in frequency for several shapes. Additionally, we obtain better reconstruction accuracy with up to a 79% decrease in the PN RMSE. While previously, feasible trap trajectories needed to be tuned manually for each shape and levitator, our approach requires calibration of each individual levitator just once. We are confident that the ideas in this article could form the basis for future content authoring tools for acoustic levitation displays and bring them a key step closer to real-world applications.

REFERENCES

- Marco A. B. Andrade, Anne L. Bernassau, and Julio C. Adamowski. 2016. Acoustic levitation of a large solid sphere. *Applied Physics Letters* 109, 4 (2016), 044101. Retrieved from <https://doi.org/10.1063/1.4959862>.
- Myroslav Bachynskyi, Viktorija Paneva, and Jörg Müller. 2018. LeviCursor: Dexterous interaction with a levitating object. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. ACM, New York, NY, 253–262. DOI: <https://doi.org/10.1145/3279778.3279802>
- Hans Georg Bock and Karl-Josef Plitt. 1984. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes* 17, 2 (1984), 1603–1608.
- Richard W. Bowen, Jordan Pola, and Leonard Martin. 1974. Visual persistence: Effects of flash luminance, duration and energy. *Vision Research* 14, 4 (1974), 295–303. DOI: [https://doi.org/10.1016/0042-6989\(74\)90079-0](https://doi.org/10.1016/0042-6989(74)90079-0)
- Henrik Bruus. 2012. Acoustofluidics 7: The acoustic radiation force on small particles. *Lab Chip* 12, 6 (2012), 1014–1021. DOI: <https://doi.org/10.1039/C2LC21068A>
- John Charles Butcher. 2016. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons.
- Timm Faulwasser. 2012. *Optimization-based solutions to constrained trajectory-tracking and path-following problems*. Ph.D. Dissertation. Otto-von-Guericke-Universität Magdeburg, Universitätspl. 2.
- Timm Faulwasser, Veit Hagenmeyer, and Rolf Findeisen. 2014. Constrained reachability and trajectory generation for flat systems. *Automatica* 50, 4 (2014), 1151–1159. DOI: <https://doi.org/10.1016/j.automatica.2014.02.011>
- Timm Faulwasser, Tobias Weber, Pablo Zometa, and Rolf Findeisen. 2017. Implementation of nonlinear model predictive path-following control for an industrial robot. *IEEE Transactions on Control Systems Technology* 25, 4 (2017), 1505–1511. DOI: <https://doi.org/10.1109/TCST.2016.2601624>
- Andreas Rene Fender, Diego Martinez Plasencia, and Sriram Subramanian. 2021. ArticLev: An integrated self-assembly pipeline for articulated multi-bead levitation primitives. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY. DOI: <https://doi.org/10.1145/3411764.3445342>
- Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. 1995. Flatness and defect of non-linear systems: Introductory theory and examples. *International Journal of Control* 61, 6 (1995), 1327–1361.
- Daniele Foresti, Majid Nabavi, Mirko Klingauf, Aldo Ferrari, and Dimos Poulidakos. 2013. Acoustophoretic contactless transport and handling of matter in air. *Proceedings of the National Academy of Sciences* 110, 31 (2013), 12549–12554. DOI: <https://doi.org/10.1073/pnas.1301860110> arXiv:<https://www.pnas.org/content/110/31/12549.full.pdf>
- Euan Freeman, Julie Williamson, Sriram Subramanian, and Stephen Brewster. 2018. Point-and-Shake: Selecting from levitating object displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY. DOI: <https://doi.org/10.1145/3173574.3173592>
- Tatsuki Fushimi, Bruce W. Drinkwater, and Thomas L. Hill. 2020. What is the ultimate capability of acoustophoretic volumetric displays? *Applied Physics Letters* 116, 24 (2020), 244101. DOI: <https://doi.org/10.1063/5.0008351>
- Tatsuki Fushimi, Bruce W. Drinkwater, Asier Marzo, and Thomas L. Hill. 2018. Non-linear trapping stiffness of mid-air single-axis acoustic levitators. *Applied Physics Letters* 113, 3 (2018), 034102. DOI: <https://doi.org/10.1063/1.5034116>
- Tatsuki Fushimi, Asier Marzo, Bruce W. Drinkwater, and Thomas L. Hill. 2019. Acoustophoretic volumetric displays using a fast-moving levitated particle. *Applied Physics Letters* 115, 6 (2019), 064101. DOI: <https://doi.org/10.1063/1.5113467>
- Thomas Geijtenbeek and Nicolas Pronost. 2012. Interactive character animation using simulated physics: A state-of-the-art review. *Computer Graphics Forum* 31, 8 (2012), 2492–2515. DOI: <https://doi.org/10.1111/j.1467-8659.2012.03189.x>
- Ryuji Hirayama, Diego Martinez Plasencia, Nobuyuki Masuda, and Sriram Subramanian. 2019. A volumetric display for visual, tactile and audio presentation using acoustic trapping. *Nature* 575, 7782 (2019), 320–323.
- Seki Inoue, Shinichi Mogami, Tomohiro Ichihara, Akihito Noda, Yasutoshi Makino, and Hiroyuki Shinoda. 2019. Acoustical boundary hologram for macroscopic rigid-body levitation. *The Journal of the Acoustical Society of America* 145, 1 (2019), 328–337.
- Kota Kumagai, Shun Miura, and Yoshio Hayasaki. 2021. Colour volumetric display based on holographic-laser-excited graphics using drawing space separation. *Scientific Reports* 11, 1 (2021), 1–9.
- Olvi L. Mangasarian. 1994. *Nonlinear Programming*. SIAM.
- Asier Marzo and Bruce W. Drinkwater. 2019. Holographic acoustic tweezers. *Proceedings of the National Academy of Sciences* 116, 1, 84–89. DOI: <https://doi.org/10.1073/pnas.1813047115>
- Asier Marzo, Sue Ann Seah, Bruce W. Drinkwater, Deepak Ranjan Sahoo, Benjamin Long, and Sriram Subramanian. 2015. Holographic acoustic elements for manipulation of levitated objects. *Nature Communications* 6, 1 (2015), 8661. <http://dx.doi.org/10.1038/ncomms9661>
- Rafael Morales, Iñigo Ezcurdia, Josu Irisarri, Marco A. B. Andrade, and Asier Marzo. 2021. Generating airborne ultrasonic amplitude patterns using an open hardware phased array. *Applied Sciences* 11, 7 (2021). DOI: <https://doi.org/10.3390/app11072981>
- Rafael Morales, Asier Marzo, Sriram Subramanian, and Diego Martínez. 2019. Levi-Props: Animating levitated optimized fabric structures using holographic acoustic tweezers. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, 651–661. DOI: <https://doi.org/10.1145/3332165.3347882>
- Tobias Nägele, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. 2017. Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics* 36, 4 (2017), 1–10.
- Christopher Nielsen and Manfredi Maggiore. 2008. On local transverse feedback linearization. *SIAM Journal on Control and Optimization* 47, 5 (2008), 2227–2250.
- Mohd Adili Norasikin, Diego Martinez-Plasencia, Gianluca Memoli, and Sriram Subramanian. 2019. SonicSpray: A technique to reconfigure permeable mid-air displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. 113–122.
- Mohd Adili Norasikin, Diego Martinez Plasencia, Spyros Polychronopoulos, Gianluca Memoli, Yutaka Tokuda, and Sriram Subramanian. 2018. SoundBender: Dynamic acoustic control behind obstacles. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, 247–259. DOI: <https://doi.org/10.1145/3242587.3242590>
- Yoichi Ochiai, Takayuki Hoshi, and Jun Rekimoto. 2014. Pixie dust: Graphics generated by levitated and animated objects in computational acoustic-potential field. *ACM Transactions on Graphics* 33, 4 (2014), 13 pages. DOI: <https://doi.org/10.1145/2601097.2601118>
- Themis Omirou, Asier Marzo, Sue Ann Seah, and Sriram Subramanian. 2015. LeviPath: Modular acoustic levitation for 3D path visualisations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, 309–312. DOI: <https://doi.org/10.1145/2702123.2702333>
- Themis Omirou, Marzo Asier Perez, Sriram Subramanian, and Anne Roudaut. 2016. Floating charts: Data plotting using free-floating acoustically levitated representations. In *Proceedings of the 2016 IEEE Symposium on 3D User Interfaces*. 187–190. DOI: <https://doi.org/10.1109/3DUI.2016.7460051>
- Viktorija Paneva, Myroslav Bachynskyi, and Jörg Müller. 2020. Levitation simulator: Prototyping for ultrasonic levitation interfaces in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY.
- Diego Martinez Plasencia, Ryuji Hirayama, Roberto Montano-Murillo, and Sriram Subramanian. 2020. GS-PAT: High-speed multi-point sound-fields for phased arrays of transducers. *ACM Transactions on Graphics* 39, 4 (2020), 12 pages. DOI: <https://doi.org/10.1145/3386569.3392492>
- Maxime Reynal, Euan Freeman, and Stephen Brewster. 2020. Avoiding collisions when interacting with levitating particle displays. In *Proceedings of the Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, 1–7. DOI: <https://doi.org/10.1145/3334480.3382965>
- Mike Roberts and Pat Hanrahan. 2016. Generating dynamically feasible trajectories for quadrotor cameras. *ACM Transactions on Graphics* 35, 4 (2016), 11 pages. DOI: <https://doi.org/10.1145/2897824.2925980>
- Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F. Cohen. 1996. Efficient generation of motion transitions using spacetime constraints. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. Association for Computing Machinery, New York, NY, 147–154. DOI: <https://doi.org/10.1145/237170.237229>
- Deepak Ranjan Sahoo, Takuto Nakamura, Asier Marzo, Themis Omirou, Michihiro Asakawa, and Sriram Subramanian. 2016. JOLED: A mid-air display based on

- electrostatic rotation of levitated janus objects (*UIST'16*). In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, 437–448. DOI : <https://doi.org/10.1145/2984511.2984549>
- Kang Shin and Neil McKay. 1985. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control* 30, 6 (1985), 531–541.
- D. E. Smalley, E. Nygaard, K. Squire, J. Van Wagoner, J. Rasmussen, S. Gneiting, K. Qaderi, J. Goodsell, W. Rogers, M. Lindsey, et al. 2018. A photophoretic-trap volumetric display. *Nature* 553, 7689 (2018), 486.
- Walter Le Conte Stevens. 1899. *A Text-Book of Physics–Sound*. American Association for the Advancement of Science.
- Diederik Verscheure. 2009. *Contribution to Contact Modeling and Identification and Optimal Robot Motion Planning*. Ph. D. Dissertation. Katholieke Universiteit Leuven, Faculteit Ingenieurswetenschappen, Department Werktuigkunde.
- Oskar Von Stryk. 1993. Numerical solution of optimal control problems by direct collocation. In *Proceedings of the Optimal Control*. Springer, 129–143.
- Andrew Witkin and Michael Kass. 1988. Spacetime constraints. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*. Association for Computing Machinery, New York, NY, 159–168. DOI : <https://doi.org/10.1145/54852.378507>

Received 25 August 2021; revised 8 February 2022; accepted 10 February 2022