

A Trajectory Planning Model for the Manipulation of Particles in Microfluidics

L. MEACCI, F.F. ROCHA, A.A. SILVA, P.V. PRAMIU and G.C. BUSCAGLIA

Received on October 03, 2017 / Accepted on May 30, 2018

ABSTRACT. Many important microfluidic applications require the control and transport of particles immersed in a fluid. We propose a model for automatically planning good trajectories from an arbitrary point to a target in the presence of obstacles. It can be used for the manipulation of particles using actuators of mechanical or electrical type. We present the mathematical formulation of the model and a numerical method based on the optimization of travel time through the Bellman's principle. The implementation is focused on square grids such as those built from pixelated images. Numerical simulations show that the trajectory tree produced by the algorithm successfully avoids obstacles and stagnant regions of the fluid domain.

Keywords: Trajectory planning, manipulation of particles in microfluidics, Bellman's principle.

1 INTRODUCTION

Particle manipulation is an important topic in microfluidics, a science that studies the behavior of small-scale fluids. A growing interest exists in the steering of particles in microfluidic systems, especially in the biological and medical fields, to analyze elements of very small size such as proteins or DNA. In particular, improving the ability to manipulate and to steer particles in microfluidics can allow to better examine, control and treat biological materials. Some interesting examples are the microfluidic device presented in [13] or the use of optical tweezers as tools to move in a controlled way individual particles, as shown in [6].

The control of the trajectory of particles immersed in a fluid by boundary actuators is the topic presented in the articles by Chaudhary and Shapiro [4] and Tuval et. al. [12]. The microfluidic modeling in these works proposes the displacement of the particles from one point to another by a pre-established trajectory. The idea is to identify the location of the particle in real time so that a control algorithm calculates the demand on each actuator for the flow to carry the particle along the desired path. In Armani et. al., [1] and [2], some examples can be found of the use of control

*Corresponding author: Luca Meacci – E-mail: luca.meacci@gmail.com
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Av. Trabalhador São-carlense, 400, 13566-590, São Carlos, SP, Brasil. E-mail: luca.meacci@usp.br; fr.franciane@usp.br; ariane@usp.br; ppramiu@usp.br; gustavo.buscaglia@icmc.usp.br

techniques. A generalization for the manipulation of particles in the three-dimensional case also exists in the study of Probst and Shapiro [10].

In all published methods discussed above, a predefined trajectory is necessary as a datum but no mention is given about how to build such a trajectory. In fact, in their examples the trajectory is introduced manually. The present work proposes a method to automatically define a “good” trajectory γ^* that can then be used as input for those particle manipulation techniques. By “good” we intend that γ^* not only avoids obstacles, but also avoids parts of the fluid domain where the actuators are ineffective at producing flow (stagnant regions, or regions of low controllability). These requirements are fulfilled by time-optimal trajectories, which are the ones proposed here. In fact, we propose an algorithm that works on a rectangular grid and could be implemented directly on a pixelated image of the domain. The trajectories are discrete in the sense that they are broken lines joining neighbor grid points. This is a very practical, though not smooth, parameterization of possible paths. The resulting trajectory γ^* can eventually be smoothed out before passing it to the manipulation algorithm. The combination of the proposed method with the manipulation techniques of Shapiro and coworkers ([4], [1], [2] and [10]) would allow for an integrated system of detection, planned manipulation and control of particles towards an arbitrary target.

The presentation begins by describing the general physical model and recalling the techniques used for particle steering when the desired trajectory is known. This allows us to introduce several concepts needed for trajectory planning. Then the continuous problem of optimizing trajectories (Bellman’s principle) with respect to arrival time is described, together with the proposed discretization. Finally, applications are reported in the section of numerical results. They show the good performance of the algorithm in finding suitable, sometimes counterintuitive, trajectories.

2 THE PHYSICAL MODEL

We study the manipulation of a particle in a microfluidic device in the two-dimensional case, the extension to 3D being immediate. In fact, we can see the 2D case applicable to flows between flat plates parallel to the plane xy , in the case when the plates are sufficiently close.

Let us consider therefore a square domain $(0, L)^2 = \Omega$, for the flow of a fluid without inertia with the presence of obstacles, walls and also of actuators. As we can see in the Figure 1, the actuators act in some regions of the boundary of the domain $\Gamma_N, \Gamma_S, \Gamma_E$ and Γ_W , being placed on the respective north, south, east and west borders.

At each actuator we apply a value $X^N, X^S, X^E \in X^W$. Since the driving force of the flow is the difference between actuator values (pressure or electrical potential differences), we can fix X^E to zero without losing generality and work only with the variables $\underline{X} = (X^N, X^S, X^W)^T$. Inertial effects are generally negligible in microfluidics. As a consequence, each $\underline{X} \in \mathcal{A}$ (where \mathcal{A} is the set of admissible actuator values) leads to a velocity field according to the following linear relation

$$\vec{v}(\vec{x}, \underline{X}) = \underline{B}(\vec{x})\underline{X}, \quad \vec{x} \in \Omega. \quad (2.1)$$

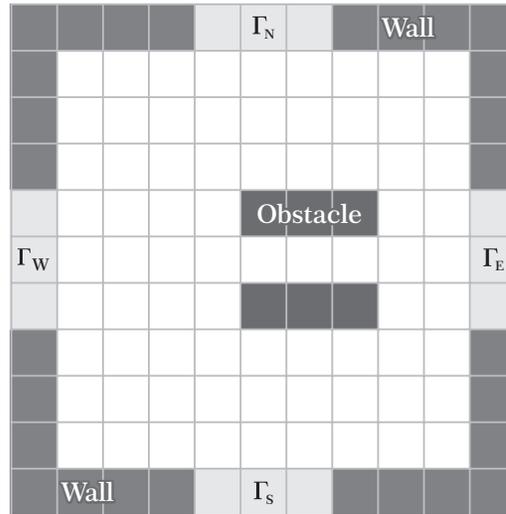


Figure 1: Sketch of the domain configuration.

The matrix $\underline{B}(\vec{x}) = (B^N(\vec{x}), B^S(\vec{x}), B^W(\vec{x}))^T$ contains as columns the components of the velocity field at the point \vec{x} with \underline{X} belonging to the canonical basis, i.e. $X^N = 1, X^S = 0$ and $X^W = 0$ for the first column; $X^N = 0, X^S = 1$ and $X^W = 1$ for the second one and $X^N = 0, X^S = 0$ and $X^W = 1$ for the third one. The Figure 2 shows an example of the velocity field, for some points of the domain, generated by imposing the canonical basis.

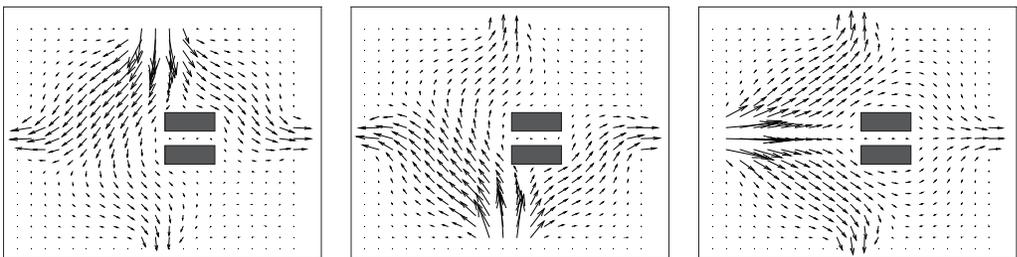


Figure 2: Velocity fields generated by imposing the canonical basis. All possible velocity fields are linear combinations of these three.

The admissible set of the controls \mathcal{A} generates an admissible set of velocities $\mathbb{V}(\vec{x})$ on each point, defined by

$$\mathbb{V}(\vec{x}) = \{ \vec{w} \in \Omega \mid \exists \underline{X} \in \mathcal{A}, \vec{w} = \underline{B}(\vec{x})\underline{X} \} .$$

Let P be a passive point particle whose initial position is \vec{x}^0 . Each control function $\underline{X}(t) : [0, T] \rightarrow \mathcal{A}$ moves P along a trajectory $\vec{r}(t)$ that is the solution of the following problem

$$\begin{cases} \frac{d\vec{r}}{dt} = \vec{v}(\vec{r}(t), \underline{X}(t)) \\ \vec{r}(0) = \vec{x}_0 . \end{cases} \tag{2.2}$$

The goal of this work is to answer the question of how to determine the control function $\underline{X}^*(t)$ that moves the particle P from the position \vec{x}^0 to a target set $Z \subset \Omega$. The target set Z may be a region of the domain where the particle is analyzed or subjected to some biochemical reaction, for example. In order to completely achieve this aim we have first to solve the problem of the particle manipulation through a predefined path. As explained in the following subsection, this approach introduces an optimization problem for maximizing the possible velocity. The solution of this preliminary problem allow us to have all the tools for achieving the main goal, which is to determine a “good” trajectory that consider the arrival time optimization.

2.1 Particle manipulation along a predefined path

Let us consider the determination of the control function $\underline{X}^*(t)$ when the path to move the particle P from \vec{x}^0 to Z is given. Let γ be the chosen path that is a smooth curve in Ω joining \vec{x}^0 to $\vec{x}^f \in Z$. For every point $\vec{x} \in \gamma$, let $\check{\tau}(\vec{x})$ be the unit vector tangent to γ at \vec{x} . In order to move P along γ it is necessary and sufficient that $\underline{B}(\vec{r}(t))\underline{X}(t)$ is parallel to $\check{\tau}(\vec{r}(t))$ for all t . Since the relation (2.1) does not depend explicitly on t , time becomes a parameter and will be omitted in what follows for conciseness. We have the following characterization:

Proposition 1. *At a point \vec{x} the control \underline{X} generates a velocity vector $\vec{w} = \vec{v}(\vec{x}, X)$ parallel to an arbitrary direction \check{d} , if and only if*

$$\underline{M}(\vec{x}, \check{d})\underline{X} \stackrel{def}{=} (\mathbb{I} - \underline{d}\underline{d}^T)\underline{B}(\vec{x})\underline{X} = 0 ,$$

where \underline{d} is the (column) array representation of \check{d} . Being $\underline{B}(\vec{x})$ of full rank, and the number of controls equal to or greater than the number of spatial dimensions, the above equation has as its solution a non-trivial subspace $S(\vec{x}, \check{d})$ of \mathbb{R}^n .

In particular, the maximum velocity that can be generated along direction \check{d} is

$$V(\vec{x}, \check{d}) = \max_{\underline{X} \in S(\vec{x}, \check{d}) \cap \mathcal{A}} \underline{d}^T \underline{B}(\vec{x}) \underline{X} ,$$

and the maximizing argument $\hat{\underline{X}}(\vec{x}, \check{d})$ is the control value that realizes the velocity $V(\vec{x}, \check{d})\check{d}$.

If the admissible set of controls is a box with minimum \underline{X}^- and maximum \underline{X}^+ values for each control variable, the following classical linear optimization problem results:

Linear optimization problem: Maximize $\underline{c}^T \underline{X}$ over $\underline{X} \in \mathbb{R}^n$ (being $\underline{c}^T = \underline{d}^T \underline{B}(\vec{x})$), with the restrictions

$$\begin{aligned} \underline{M}(\vec{x}, \check{d})\underline{X} &= 0 \\ \underline{X} &\geq \underline{X}^- \\ \underline{X} &\leq \underline{X}^+ . \end{aligned} \tag{2.3}$$

The maximum reached is the maximum velocity $V(\vec{x}, \check{d})\check{d}$, and the maximizing argument is $\hat{\underline{X}}(\vec{x}, \check{d})$.

With these elements, bringing back the time variable, the problem (2.2) becomes

$$\begin{cases} \frac{d\vec{r}}{dt} = V(\vec{r}(t), \check{\tau}(\vec{r}(t))) \check{\tau}(\vec{r}(t)), \\ \vec{r}(0) = \vec{x}_0, \end{cases}$$

which will necessarily follow the curve γ at maximum speed, and $\vec{r}(t)$ has been calculated the control function arises from

$$\underline{X}^*(t) = \hat{X}(\vec{r}(t), \check{\tau}(\vec{r}(t))). \quad (2.4)$$

The control at every instant is uniquely determined by the position of the particle and the direction in which it is desired to move it. The problem of microfluidically controlling a particle to follow a predefined trajectory was addressed by Chaudhary and Shapiro, [4] and the work of Mathai et al. [8], where the simplest constraints ($\|\underline{X}\| \leq 1$, where $\|\cdot\|$ is the Euclidean norm) are considered.

Remark 2.1. In (2.4) the vector \hat{X} can be multiplied by an arbitrary $\alpha(t) \in (0, 1]$ if the desired speed of the particle is not the maximum attainable one at each point. For example, let

$$V_{\min}(\gamma) = \min_{\vec{x} \in \gamma} V(\vec{x}, \check{\tau}(\vec{x})).$$

If $V_{\min}(\gamma) = 0$, it means that it is impossible to reach the target Z in finite time along the path γ . Otherwise ($V_{\min}(\gamma) > 0$), for any $V^* \leq V_{\min}(\gamma)$ the control function that moves P along γ at constant speed V^* is obtained by first solving

$$\begin{cases} \frac{d\vec{r}}{dt} = V^* \check{\tau}(\vec{r}(t)), \\ \vec{r}(0) = \vec{x}_0, \end{cases}$$

and then setting

$$\underline{X}^*(t) = \frac{V_{\min}(\gamma)}{V(\vec{r}(t), \check{\tau}(\vec{r}(t)))} \hat{X}(\vec{r}(t), \check{\tau}(\vec{r}(t))).$$

2.2 Determining the path via arrival time optimization

When the path is not pre-defined, it can be chosen in several ways. Here we study the trajectory planning between \vec{x}^0 and the Z target by minimizing the time of arrival. Other possibilities include minimizing path length, energy expenditure, and in general any cost function depending on the path and possibly on the associated control function.

For a point $\vec{x} \in \Omega$ (controllable, i.e., such that there exists at least one function $\underline{X}(t)$ which moves P from \vec{x} to Z) we can consider the function $T(\vec{x})$ defined as the minimum time required to take P from \vec{x} to Z . The path associated with $T(\vec{x})$ is an optimal trajectory because it moves P to Z in minimum time and defines, at each point, the optimal direction $\check{\tau}(\vec{x})$ (tangent to the optimal trajectory).

Bellman's principle states that if a point \vec{y} is placed in the optimal path γ^* of \vec{x} , then the optimal trajectory from \vec{x} to \vec{y} is the arc of γ^* joining these two points, and the optimal trajectory of \vec{y} to

the target Z is the final arc of γ^* . With this principle it is possible to obtain the Hamilton-Jacobi-Bellman equation for T , according to [3]:

$$V(\vec{x}, \check{t}(\vec{x})) \check{t}(\vec{x}) \cdot \nabla T + 1 = 0 ,$$

where $\check{t}(\vec{x})$ is given by $\check{t}(\vec{x}) = \arg \min_{\|\check{d}\|=1} V(\vec{x}, \check{d}) \check{d} \cdot \nabla T(\vec{x})$. We have that the associated Hamiltonian is $H(\vec{x}, \vec{p}) = \min_{\|\check{d}\|=1} V(\vec{x}, \check{d}) \check{d} \cdot \vec{p}$, which gives the equation in the usual form (see [11])

$$H(\vec{x}, \nabla T(\vec{x})) + 1 = 0 .$$

The fundamental property we will use in the numerical treatment of this equation is the following (a direct consequence of Bellman’s principle):

Proposition 2. *Let $\vec{x} \in \Omega$, and C a closed curve (surface in 3D) such that \vec{x} is internal to C and the target Z is external to C . For each $\vec{y} \in C$, let $\zeta(\vec{y})$ be the minimum time required to carry a particle from \vec{x} to \vec{y} . Then,*

$$T(\vec{x}) = \min_{\vec{y} \in C} \{T(\vec{y}) + \zeta(\vec{y})\} . \tag{2.5}$$

Moreover, if \vec{y}^* is a point of C where the minimum is attained, \vec{y}^* is the intersection of the optimal path γ^* with the curve C .

If the curve C is small enough (contained in a ball of ratio $\rho \ll 1$), we can consider just straight trajectories between \vec{x} and C , which allows to approximate the equation (2.5), according to [3], by

$$T(\vec{x}) \approx \min_{\vec{y} \in C} \left\{ T(\vec{y}) + \frac{\|\vec{y} - \vec{x}\|}{V\left(\vec{x}, \frac{\vec{y} - \vec{x}}{\|\vec{y} - \vec{x}\|}\right)} \right\} \tag{2.6}$$

and the optimal trajectory, locally, is given by the segment $\overline{\vec{x}\vec{y}^*}$.

3 CALCULATION OF $V(\vec{X}, \check{D})$

To find the set of possible maximum velocities $V(\vec{x}, \check{d})$ at a point \vec{x} , for each given direction \check{d} , we have first to determine the admissible set of velocities $\mathbb{V}(\vec{x})$, according to the equation (2.1). We consider that the system actuators can be mechanical (imposing pressure) or electrical (imposing electrode voltage).

When the velocity field is generated by pressure sources, the governing equation for the flow is the Stokes equation, that in our setting defines the following boundary value problem

$$\begin{cases} -\mu \nabla^2 \vec{v} + \nabla p = 0 & \text{in } \Omega , \\ \nabla \cdot \vec{v} = 0 & \text{in } \Omega , \\ \sigma(p, \nabla \vec{v}) \cdot \vec{n} = -X^i & \text{on } \Gamma_i, \text{ where } i \in \{N, S, E, W\} , \\ \vec{v} = 0 & \text{on } \partial\Omega \setminus \cup_i \Gamma_i , \end{cases} \tag{3.1}$$

where $\sigma(p, \nabla \vec{v}) = -p\mathbb{I} + \mu(\nabla \vec{v} + \nabla \vec{v}^T)$ is the stress tensor and \vec{n} the unit normal to the boundary. We remark that the boundary conditions for this problem are essentially that pressure is imposed on the actuators and null velocity is imposed on the obstacles and walls.

When the system actuators are electrodes, the electric field is obtained from the electric potential Φ (or voltage) that solves the following problem

$$\begin{cases} \nabla^2 \Phi = 0 & \text{in } \Omega, \\ \vec{E} = -\nabla \Phi & \text{in } \Omega, \\ \Phi = X^i & \text{on } \Gamma_i, \text{ where } i \in \{N, S, E, W\}, \\ \vec{E} \cdot \vec{n} = 0 & \text{on } \partial\Omega \setminus \cup_i \Gamma_i. \end{cases} \quad (3.2)$$

The electric field induces a velocity field according to the relation

$$\vec{v} = m_{eo} \vec{E},$$

where m_{eo} is the electrosmotic constant relative to the interaction between the fluid and the wall [7]. In this case, Dirichlet boundary conditions for the voltage are imposed on the actuators, and Neumann homogeneous ones on the obstacles and walls.

A detailed explanation of the numerical resolution of these problems, according to MAC (Marker and Cell) discretization, is exposed in [9].

Remark 3.1. In fact, one could easily consider a combination of mechanical and electrical actuators. In this case, if a control variable X_i corresponds to a mechanical actuator, the i -th column of matrix B in (2.1) would be the velocity field obtained by solving (3.1), while if it corresponds to an electrical actuator the equation to be solved would be (3.2). We have not considered problems with both kinds of actuators simultaneously to keep the exposition simpler.

The calculation of $V(\vec{x}, \vec{d})$ involves a resolution of the linear optimization problem of the Proposition 1 on all mesh nodes for a directions set D defined by the user.

If v_1, v_2 and v_3 are the three vectors (column) calculated with the hydrodynamic solver on the node \vec{x} , and d is a direction belonging to D , the following Octave code calculates $V(\vec{x}, \vec{d})$:

```
Xmin=[-1 -1 -1];
Xmax=[1 1 1];
B=[v1, v2, v3];
c=-d'*B;
M=(eye(2,2)-d*d')*B;
[xopt fmin]=glpk(c,M,zeros(2,1),Xmin,Xmax);
V=-fmin;
```

It is important to justify that the use of the negative sign in $\underline{c} = -\underline{d}^T \underline{B}$ is because by pattern the function `glpk` minimizes instead of maximizing. We remark that the values \underline{X}^- and \underline{X}^+ are in the vectors $Xmin$ and $Xmax$. The value of the control to generate the velocity V in the direction d is given by `xopt`.

The Figure 3 (a) illustrates an example of the set $V(\vec{x}, \vec{d})$ obtained initially with the vectors

$$v1=[1;1.1]; \quad v2=[-0.2;0.6]; \quad v3=[-0.21;-1.1];$$

In this Figure 3 (a), the external polygon was generated using as minimum and maximum values the vectors

$$Xmin=[-1,-1,-1]; \quad Xmax=[1,1,1];$$

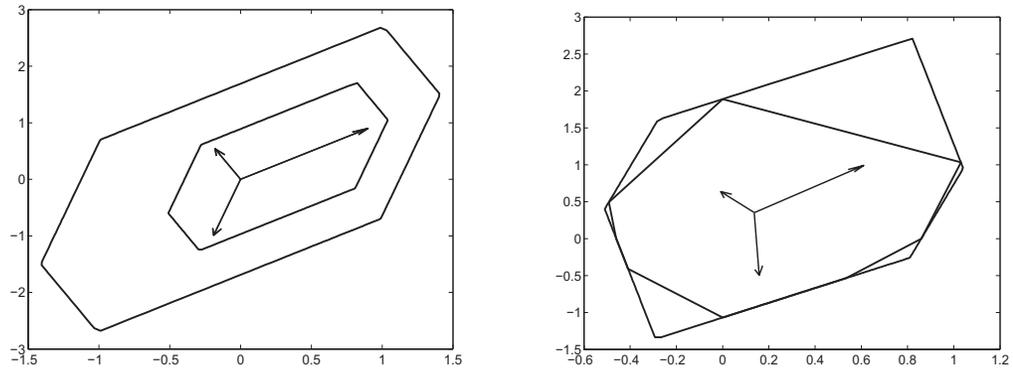
When we change the vector of minimum values of the control variables to

$$Xmin=[-0.1,-0.1,-0.1];$$

we obtain a new set $V(\vec{x}, \vec{d})$, which corresponds to the internal polygon. Notice how the polygon that represents $V(\vec{x}, \vec{d})$ loses its symmetry about the origin when $Xmin$ is different from $-Xmax$. We remark that the number of directions used to determine the region can influence the quality of the set of possible velocities generated. The Figure 3 (b) illustrates a set $V(\vec{x}, \vec{d})$ given by the values

$$v1=[1;1.1]; \quad v2=[-0.2;1.6]; \quad v3=[-0.21;-1.1];$$

where the internal polygon was generated using 8 directions and the external polygon was generated using 320 directions.



(a) With two different choices of minimum values of the control variables (see text).

With two different choices of the number of directions considered: 8 and 320.

Figure 3: Polygons representing the set $V(\vec{x}, \vec{d})$.

By performing the same procedure for all nodes of the mesh, we obtain the data necessary to calculate the optimal trajectory according to the Hamilton-Jacobi-Bellman equation. The Figure 4 shows the set $V(\vec{x}, \vec{d})$ for some points of the domain, where in the Figure 4 (a) the actuators

are of the pressure type, and in the Figure 4 (b) the actuators are of the electrode type. We can see the variability of $V(\vec{x}, \check{d})$, which is highly anisotropic and highly dependent on the geometric configuration as well as on the nature of the actuators. The regions where the polygons are smaller correspond to stagnant regions, i.e., regions where the velocity generated by the actuators hardly moves the particles at all. The regions where the polygons are bigger correspond to regions where the particles are easily transported. A “good” trajectory for manipulation avoids not just the obstacles but also these stagnant regions as much as possible.

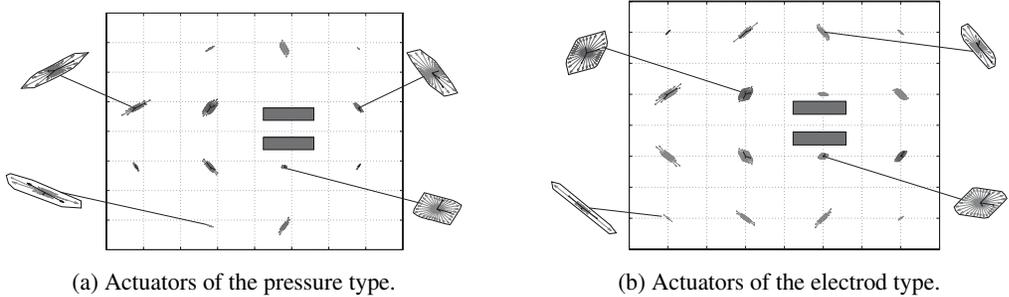


Figure 4: Representation of the set $V(\vec{x}, \check{d})$ of the possible maximum velocities (in each of the 32 directions considered) in some points of the domain.

4 OPTIMIZING TRAJECTORIES BY BELLMAN’S PRINCIPLE

The optimal trajectory, in terms of time, for moving a particle from a point \vec{x} to a target point, solves the Hamilton-Jacobi-Bellman equation. A semi-Lagrangian discretization is considered for the function T . The equation (2.6) is approximated at each node (i, j) , taking the curve C (denoted as $C_{i,j}$) as the boundary of the set of the four cells that share node (i, j) . The coordinates of node (i, j) are $\vec{x} = (X_{i-1}, Y_{j-1})$ (see Figure 5). For each direction \check{d} , the trajectory from \vec{x} to $C_{i,j}$ is approached by a straight line from \vec{x} to $\vec{y}^d \in C_{i,j}$.

The values for the discrete approximation of $T_{i,j}$ are defined by

$$T_{i,j} = \min_{\|\check{d}\|=1} \left\{ T(\vec{y}^d) + \frac{\|\vec{x} - \vec{y}^d\|}{V(\vec{x}, \check{d})} \right\}$$

where $T(\vec{y}^d)$ is an interpolation of the T values at the nodes that are placed on $C_{i,j}$. In the case of the direction \check{d} illustrated in the Figure 5, for example, $T(\vec{y}^d)$ is calculated by linearly interpolating between $T_{i+1,j}$ e $T_{i+1,j+1}$. The direction of the optimal trajectory is that in which the minimum is attained, which is generally not parallel to ∇T because of the anisotropy of the problem.

To leave a completely discrete scheme we consider the following discretization of the set of directions D

$$D = \{ \check{d}^k = (\cos(\theta_k), \sin(\theta_k)), \theta = \frac{(k-1)\pi}{m}, k = 1, \dots, m \}.$$

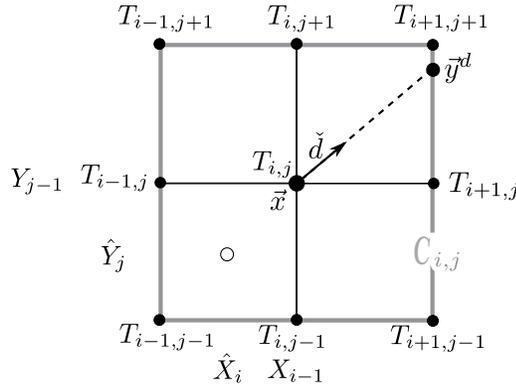


Figure 5: Trajectory, approximated by a straight line, from \vec{x} to $C_{i,j}$ for the direction \check{d} .

The algorithm requires that $V(\vec{x}, \check{d})$ has already been calculated at each vertex and has been stored as a table of values $V_{i,j}^k := V(\vec{x}_{i,j}, \check{d}^k)$, corresponding to each vertex (i, j) and to each direction \check{d}^k . $T_{i,j}$ is initialized to a large value (10^{12}) on all nodes except those that belong to the target Z , which are initialized to zero. Then we iterate all the mesh nodes that do not belong to Z updating T according with the following equation

$$T_{i,j} \leftarrow \min_{\check{d}=\check{d}^k \in D} \left\{ T(\vec{y}^{\check{d}}) + \frac{\|\vec{x} - \vec{y}^{\check{d}}\|}{V_{i,j}^k} \right\}. \tag{4.1}$$

This algorithm is discussed in [5], including error estimators for the solution in the norm of $L^\infty(\Omega)$, which is important because the exact solution in general is not smooth and can not be understood in the classical sense.

Remark 4.1 (Complexity). Calculations necessary to start the previous algorithm are:

- Computing the matrix \underline{B} according to the linear relation (2.1) for the N nodes and n actuators with cost $n \mathcal{O}(N^a)$, where $\mathcal{O}(N^a)$ is the cost of solving one hydrodynamic problem (typically $a \approx$ from 1.3 to 1.5);
- Computing $\{V_{ij}^k\}$ with cost $mN \mathcal{O}(n^b)$, obtained by the cost $\mathcal{O}(n^b)$ of solving one linear optimization problem (2.3) in dimension n (typically $b \approx 2$), for all N nodes and all m directions.

From (4.1) the cost of the HJB iteration is $\mathcal{O}(mN)$, and the number of required iterations is $\mathcal{O}(N^{1/s})$, s being the number of space dimensions. Summarizing, the total cost adds up to $\mathcal{O}(nN^a + mNn^b + mN^{1+1/s})$. Since $a \approx 1 + 1/s$, in general the HJB iterations dominate the total cost, but if the number of actuators n is large the cost of the linear optimization problems can also be significant.

5 RESULTS

In this section, we consider the set D generated by 8 directions. The accuracy of the calculation depends on the number of the considered directions, and so does the computational cost. The choice of 8 directions has the advantage of not requiring the interpolation step, since the directions considered in (4.1), with $k = 1, \dots, 8$, are determined by the neighboring nodes. Therefore for each node (i, j) the optimal trajectory is a line that connects it to the neighboring mesh node at which the minimum in (4.1) is attained. As a consequence, the set of numerical trajectories produced by our algorithm has a tree structure. For this reason, to plot the numerical solution we draw all the optimal trajectories that pass through the mesh nodes, which has the geometrical appearance of a tree rooted at the target Z .

In the next simulations, we consider a domain $\Omega = \{(x_1, x_2) \in \mathbb{R}^2, 0 < x_1, x_2 < 10\}$ discretized in a grid of 40×40 nodes, except when indicated otherwise. When the flux is induced by an electrical potential we assume an electrosmotic constant $m_{eo} = 1$ while when it is generated by a pressure source we consider viscosity $\mu = 0.1$. The actuators' limit values are set to $\underline{X}^- = (-1, -1, -1)$ and $\underline{X}^+ = (1, 1, 1)$.

Figure 6 shows the trees of optimal trajectories obtained for both electrical (subfigure (a)) and mechanical (subfigure (b)) actuators.

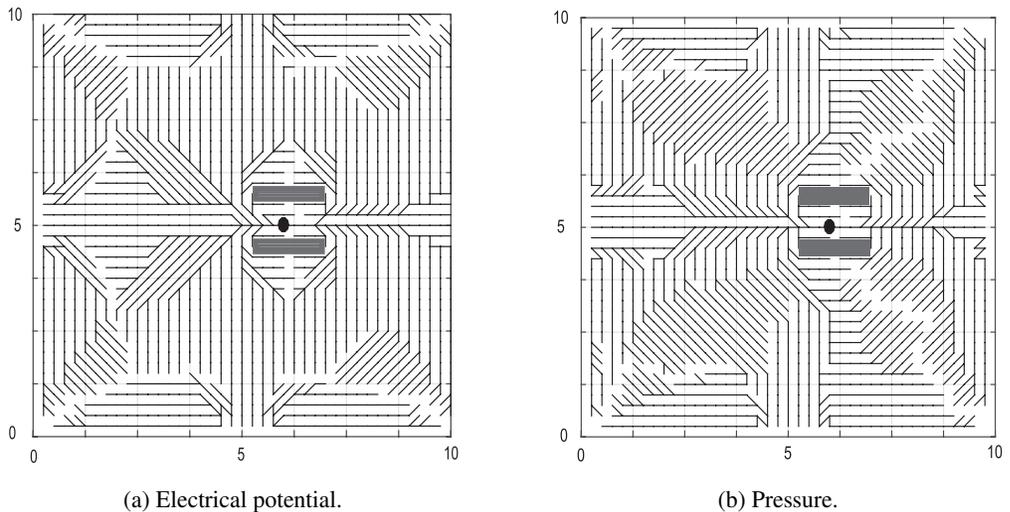


Figure 6: Optimal trajectory configuration with range of actuators between -1 and 1 . The destacked point is the arrival target.

One can notice that the trajectories between the two plates of the obstacle are essentially horizontal. This is because the actuators cannot generate significant vertical velocity there due to the presence of the plates. With this, it turns out that a particle that is between the plates but below or above Z must first take a path to get out of the space between the plates and then walk back in towards the target point. We find worthy of note that the optimal trajectories tend to circumvent

the obstacle so as to traverse regions (more distant to the obstacle) where the admissible velocities are higher. Notice also that the trajectory trees have similarities for both types of actuators, mainly differing at the obstacles' boundaries, where the mechanically-driven flow has zero velocity while the electrically-driven one only imposes the normal component of the velocity to be zero.

The model exhibits convergent behavior with respect to mesh refinement in all cases considered. As an example, in Figure 7 we compare the resulting trees for the electrically-driven model, with meshes 40×40 and 80×80 .

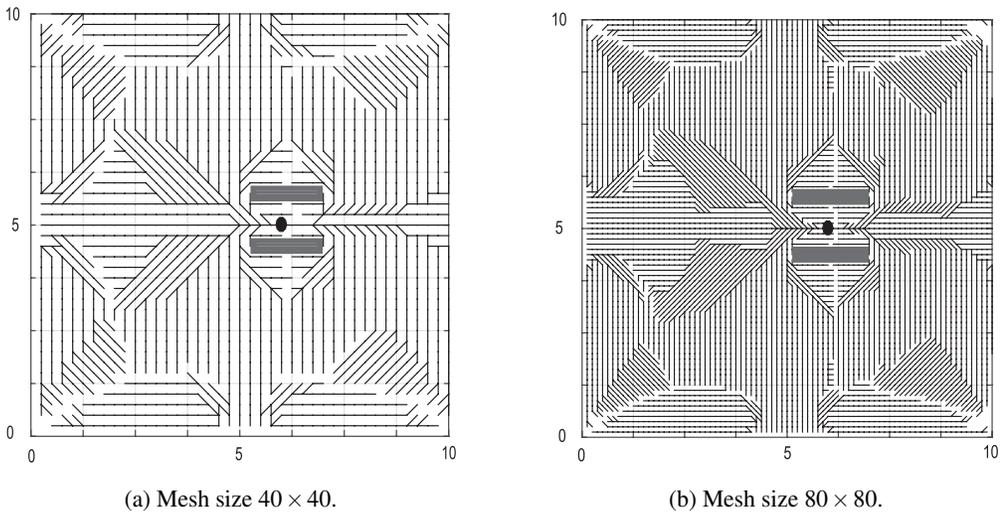


Figure 7: Effect of mesh refinement on the resulting trajectory trees, for the electrically-driven flow.

The optimal trajectories may be counterintuitive. Let us consider the flow generated by pressure actuators setting as target Z a single point close to the obstacle as shown in Figure 8. Then we consider two possible start points, labeled with the letters A and B . In the case with A as starting point, the model designs an optimal trajectory that is a straight segment. The travel time t computed by integrating the field $V(\cdot, \vec{d})$ between A and Z along a straight line ($\check{c}_1 = 1, \check{c}_2 = 0$) coincides with $t_{op} = T(A)$, where T is the solution of the discrete Hamilton-Jacobi-Bellman equation (4.1). On the other hand, when the starting point is B the optimal trajectory proposed by the model is not straight. It assumes a form that is not of immediate intuition. The designed path, although it is longer than the straight path, results in a shorter travel time $t_{op} = 6.7$ than that obtained along the straight one $t = 9.1$. This advantage is due to the greater efficiency of the actuators in the region traveled by the optimal trajectory. The values to be applied to the actuators to manipulate the particle along the proposed path are easily obtained from (2.4) and are shown in Figure 9 as functions of time for the case that has B as starting point. As usual in time-optimal control, the actuator values exhibit a bang-bang-like behavior. The geometrical path is the most

fundamental result since it avoids obstacles and stagnant regions. This path can then be traveled at non-optimal velocities, with smoother control functions, as discussed in Remark 2.1.

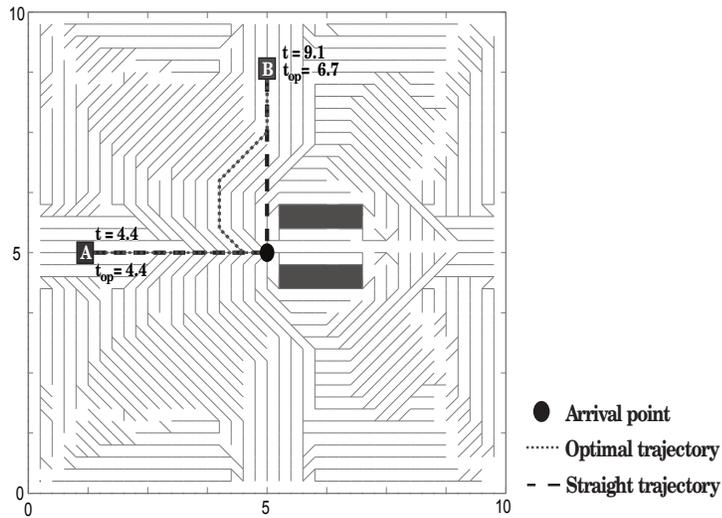


Figure 8: Comparison of the optimal trajectories with straight ones.

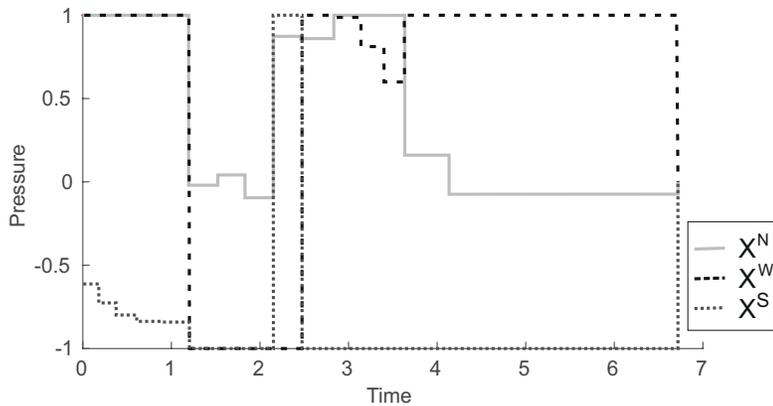


Figure 9: Actuators values over time along the optimal path starting at B .

6 CONCLUSIONS

We have presented an integrated model to identify convenient trajectories for the manipulation and control of particles in microfluidics. The achievement of the above-mentioned objective involved various mathematical problems:

- To find an admissible set of velocities $\nabla(\vec{x})$ on each point \vec{x} generated by a set of admissible actuator values \mathcal{A} , solving

- a Stokes problem for the velocity field induced by pressure sources;
- a Poisson problem for the electrical potential induced by a voltage imposed at electrodes, which in turn induces an electrosmotic velocity field;
- To determine the maximum admissible velocity $V(\vec{x}, \check{d})$ at each point \vec{x} and for each direction \check{d} , through the resolution of a linear optimization problem;
- To provide the control functions $\underline{X}^*(t)$ arising from the resolution of an initial value problem for the trajectory ODE;
- To plan the trajectory by minimizing the arrival time through the numerical implementation of Bellman's principle.

The proposed method automatically identifies a trajectory for a particle that we must manipulate from a departure point to the target in minimum time (and also provides the corresponding values to impose on the actuators). The numerical results show that the trajectories found not only avoid the obstacles, but also avoid the regions of the domain where the actuators are ineffective at producing flow. The numerical simulations show counterintuitive optimal trajectories in this kind of problem, justifying the use of mathematical models for designing automatic manipulation systems.

It should be noted that the proposed algorithm produces not just one optimal trajectory, but the whole tree of optimal paths, starting at any $x \in \Omega$. This apparent overkill is very important when applying the model under realistic conditions, in particular considering external perturbations (noise). In fact, if the sensor in the control loop detects that a perturbation has taken the particle to a position \tilde{x} outside the original optimal path, the controller can immediately switch to the path in the optimal tree that passes through \tilde{x} . In this way, and without any additional computation, the optimal tree can be used to correct perturbations so that the particle eventually reaches Z .

ACKNOWLEDGEMENTS

This work was carried out with the financial support of the CAPES and CNPq agencies. In particular, Luca Meacci acknowledges for the financial support the Programa de Excelência Acadêmica (Proex) – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES) – Finance Code PROEX-9740044/D.

RESUMO. Importantes aplicações em microfluídica requerem o controle e transporte de partículas imersas em um fluido. Propomos um modelo para o planejamento automático de boas trajetórias de um ponto arbitrário até um alvo, na presença de obstáculos. Esse modelo pode ser utilizado para a manipulação de partículas fazendo uso de atuadores do tipo mecânico ou elétrico. Apresentamos a formulação matemática do modelo e um método numérico baseado na otimização do tempo de chegada através do princípio de Bellman. A implementação considera malhas quadradas, as quais são construídas a partir de imagens

pixeladas. Simulações numéricas mostram que a árvore de trajetórias obtida do algoritmo evita obstáculos e regiões estagnadas do domínio fluídico com sucesso.

Palavras-chave: Planejamento de trajetória, manipulação de partículas em microfluídica, Princípio de Bellman.

REFERENCES

- [1] M. Armani, S. Chaudhary, R. Probst, S. Walker & B. Shapiro. Control of microfluidic systems: two examples, results, and challenges. *International Journal of Robust and Nonlinear Control*, **15**(16) (2005), 785–803.
- [2] M. Armani, Z. Cummins, J. Gong, P. Mathai, R. Probst, C. Ropp, E. Waks, S. Walker & B. Shapiro. Feedback Control of Microflows. In “Feedback Control of MEMS to Atoms”. Springer (2012), pp. 269–319.
- [3] S. Cacace, E. Cristiani & M. Falcone. Can Local Single-Pass Methods Solve Any Stationary Hamilton–Jacobi–Bellman Equation? *SIAM Journal on Scientific Computing*, **36**(2) (2014), A570–A587.
- [4] S. Chaudhary & B. Shapiro. Arbitrary steering of multiple particles independently in an electro-osmotically driven microfluidic system. *IEEE Transactions on Control Systems Technology*, **14**(4) (2006), 669–680.
- [5] E. Cristiani. “Numerical methods for optimal control problems: part II: local single-pass methods for stationary HJ equations”. IAC-CNR, Roma (2013). p. 21. Class notes.
- [6] J.E. Curtis, B.A. Koss & D.G. Grier. Dynamic holographic optical tweezers. *Optics communications*, **207**(1) (2002), 169–175.
- [7] B.J. Kirby. “Micro-and nanoscale fluid mechanics: transport in microfluidic devices”. Cambridge university press (2010).
- [8] P.P. Mathai, A.J. Berglund, J.A. Liddle & B.A. Shapiro. Simultaneous positioning and orientation of a single nano-object by flow control: theory and simulations. *New Journal of Physics*, **13**(1) (2011), 013027.
- [9] L. Meacci, F.F. Rocha, A.A. Silva, P.V. Pramiu & C.G. Buscaglia. Planejamento de trajetória para a manipulação de partículas em microfluídica. *CQD-Revista Eletrônica Paulista de Matemática*, **10** (2017), 19–37.
- [10] R. Probst & B. Shapiro. Three-dimensional electrokinetic tweezing: device design, modeling, and control algorithms. *Journal of Micromechanics and Microengineering*, **21**(2) (2011), 027004.
- [11] J.A. Sethian & A. Vladimirovsky. Ordered upwind methods for static Hamilton–Jacobi equations. *Proceedings of the National Academy of Sciences*, **98**(20) (2001), 11069–11074.
- [12] I. Tuval, I. Mezić, F. Bottausci, Y.T. Zhang, N.C. MacDonald & O. Piro. Control of particles in microelectrode devices. *Physical review letters*, **95**(23) (2005), 1–4.
- [13] Y. Zhou, Y. Wang & Q. Lin. A microfluidic device for continuous-flow magnetically controlled capture and isolation of microparticles. *Journal of Microelectromechanical Systems*, **19**(4) (2010), 743–751.