Grant Number: FA9550-15-1-0530

New Approach to the Optimal Control of Level Sets Generated by Partial Equations to Bridge the Gap Between Computational Mathematics and Control of Complex Systems

Principal Investigator: Angelo Alessandri

Period of Performance: 30 September 2015 – 29 September 2018

Final Report

Contents

List of Figures

1 Summary

We have attacked the problem of optimally driving the level sets associated with the solution of a Hamilton-Jacobi equation by relying on an approximation scheme based on the extended Ritz method. The complete description of such an approach is illustrated in Section 2. In Section **??**, we will describe an approach to the control of the normal flow equation by using either velocity field or source term. Such results are based on a different control paradigm as compared with that of Section 2. More specifically, we deal with stabilizing feedback controllers, all provided with a proof of stability.

Two experimental setups have been developed. The first one has been constructed in an indoor laboratory. The main motivation to devote to this test rig was that of making students gain experience with practice of level sets methods, as shown in Section 3. The second experimental setup is an airlift photobioreactor, in which microalgae are used to capture CO_2 from flue gas and to treat wastewater. In Section 4, we will describe and motivate the role played by level set methods in such applications.

2 Optimal Control of Level Sets

First of all, we will define the problem and prove the existence of solutions. Second, we will describe a suitable optimization method to find approximates solutions by using descent algorithms. Among such methods, we will focus on a Gauss-Newton algorithm, which can be efficiently implemented by using an extended Kalman filter (EKF). Finally, we will describe the simulation results.

Concerning the optimal control of level sets, we have attacked the problem by using the closed-loop scheme depicted in Fig. 1. Since in general it is difficult to solve such a problem, an approximation scheme based on the extended Ritz method has been investigated proposed to find suboptimal solutions. The control law is forced to take on a fixed structure that depends (in general nonlinearly) on a finite number of parameters to be suitably chosen. Therefore, we have devised an approach for the selection of the parameters by using gradient-based optimization techniques based on interior point and sequential quadratic programming methods. Toward this end, the adjoint equation is derived to compute the gradient of the cost functional with respect to the parameters of the control law.

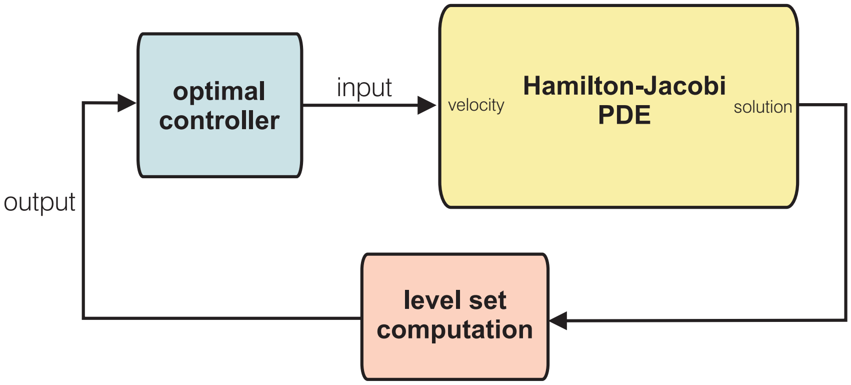

Figure 1: Sketch of the proposed optimal closed-loop control.

We have proved the existence of solutions to the considered problem by choosing suitable spaces of functions, where it is shown that the problem is well formulated under mild conditions and for which the approximation results we rely on hold.

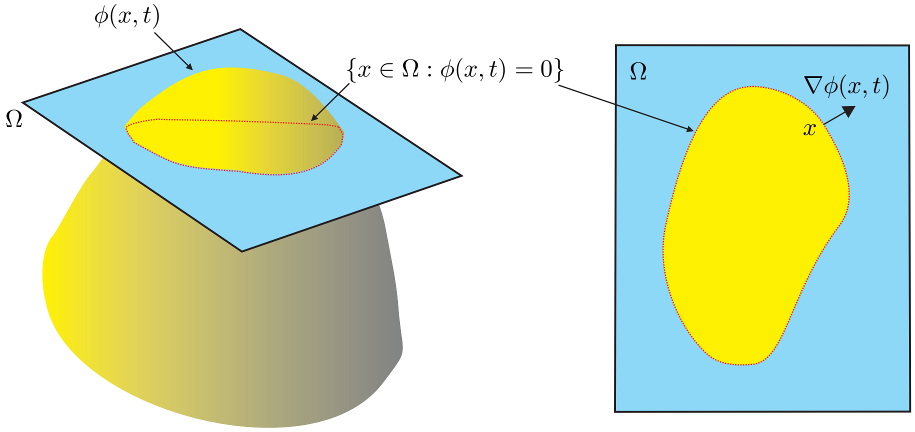

Figure 2: Examples of fronts described by zero level sets.

2.1 Optimal control of level set dynamics

Given the space domain $\Omega \subset \mathbb{R}^q$ and $t \geq 0$, level set methods consider the front implicitly represented at each time t as the zero level set of a function $\phi : \Omega \times [0,T] \to \mathbb{R}$, where $T > 0$ is a given time horizon. The front $x(t,s)$ corresponding to a level equal to c at time t is defined as the set of points such that $\phi(x(t,s),t) = c$, where s is the arc-length parameter of the initial curve $x(0,s)$, as shown in Fig. 3. If we differentiate with respect to t, we obtain the Hamilton-Jacobi equation

$$\phi_t(x,t) + v(x,t) \cdot \nabla\phi(x,t) = 0\,, \tag{1}$$

where

$$v(x,t) = \frac{d}{dt}x(t,s)$$

is the Lagrangian material particle velocity that gives the direction of propagation of the front at the point $x(t,s)$ and ∇ denotes the spatial gradient. Let us consider, for instance, the velocity $v(x,t)$ proportional to the normal to the front:

$$v(x,t) = u\,\frac{\nabla\phi(x,t)}{|\nabla\phi(x,t)|}, \tag{2}$$

where u is the propagation speed. If we replace the expression of v given by (7) in (6), we obtain the so-called normal flow equation:

$$\phi_t(x,t) + u\,|\nabla\phi(x,t)| = 0, \tag{3}$$

with initial conditions $\phi_0 : \Omega \to \mathbb{R}$ that have to be fixed, i.e., $\phi(x,0) = \phi_0(x)$, for all $x \in \Omega$ (a usual choice is to take the signed distance to the initial front).

The behavior of the function ϕ in both space and time can be manipulated by a suitable choice of u. In general, we may regard u as a control action depending on space and time, i.e., $u : \Omega \times [0,T] \to \mathbb{R}$. The level set control of (3) consists in choosing $u(x,t)$ to move the propagating front described by a level set of $\phi(x,t)$ as desired.

The evolution of the level sets of ϕ over time may be associated with a performance index that depends on the boundary or on the interior of the shape (Alessandri, Bagnerini, and Gaggero, 2014). In general, we consider the following cost measuring the performance of the control action u:

$$J(\phi,u) = \int_0^T \int_\Omega h(\phi(x,t),u(x,t),t)\,dx\,dt + \int_\Omega \bar{h}(\phi(x,T))\,dx, \tag{4}$$

where h and $\bar{h}$ are functionals defined on $\mathcal{F} \times \mathcal{U}$ and $\mathcal{F}$, respectively, with $\mathcal{F}$ and $\mathcal{U}$ denoting the sets of the admissible functions $\phi(x,t)$ and $u(x,t)$, respectively.

We search for the optimal control action $u^\circ \in \mathcal{U}$ that minimizes the cost functional $J(\phi, u)$ subject to the normal flow equation (3), i.e.,

$$u^\circ \in \underset{u \in \mathcal{U} \,\text{s.t.}\, (3) \text{ holds}}{\operatorname{argmin}} \; J(\phi, u)\,. \tag{5}$$

Unfortunately, it is almost impossible to find the exact analytic solution of (5), and therefore the search for approximate solutions is mandatory in practice. However, before attacking the solution of this problem, we need to prove that at least a solution exists, as shown in the sequel.

2.2 Existence of Solutions to the Optimal Control Problem

Let us consider a set $\Omega \subset \mathbb{R}^q$ open, bounded, and smooth and the time $t \in [0, T]$, with $T > 0$. LS methods represent a moving front or interface at each time t, i.e., a curve in two dimensions or a surface in three dimensions separating two regions, as the zero LS of a multidimensional function $\phi : \overline{\Omega} \times [0, T] \to \mathbb{R}$. The interface $x(t, s)$ is given at time t by the points such that $\phi(x(t, s), t) = 0$, where s is the arc-length parameter of the initial curve $x(0, s)$. Figure 3 displays fronts at two different time instants t_1 and t_2. By differentiating w.r.t. t, we obtain

$$\phi_t(x, t) + v(x, t) \cdot \nabla\phi(x, t) = 0 \tag{6}$$

i.e., a Hamilton-Jacobi equation, where

$$v(x, t) := \frac{d}{dt} x(t, s)$$

is the Lagrangian particle velocity giving the direction of propagation of the interface in the point $x(t, s)$ and $\nabla\phi(x, t)$ is the Fréchet gradient of $\phi(x, t)$ w.r.t. the space. If we choose $v(x, t)$ proportional to the normal to the front, i.e.,

$$v(x, t) = u\,\frac{\nabla\phi(x, t)}{|\nabla\phi(x, t)|} \tag{7}$$

where u is the speed of propagation and replace (7) in (6), we have the normal flow (NF) equation

$$\phi_t(x, t) + u(x, t)\,|\nabla\phi(x, t)| = 0 \tag{8}$$

where the speed function $u : \overline{\Omega} \times [0, T] \to \mathbb{R}$ is regarded as a control input and initial conditions $\phi_0 : \overline{\Omega} \to \mathbb{R}$, i.e., $\phi(x, 0) = \phi_0(x)$, $x \in \overline{\Omega}$. Usually, ϕ_0 is chosen as the signed distance to the initial front. Equation (8) is a Hamilton-Jacobi equation, whose solution is defined in the sense of viscosity solutions and it is based on the notion of sub- and super-differentials (Falcone and Ferretti, 2014). The $l \in \mathbb{R}$ LS of the function ϕ is a set-valued mapping $\Gamma_l : [0, T] \rightrightarrows \mathcal{C}$, where

$$\Gamma_l(t) := \{x \in \Omega : \phi(x, t) = l\}\;.$$

We deal with the problem of the optimal control of (8) for some cost functional to be minimized that provides a performance index depending on the propagating front associated with a certain LS of $\phi(x, t)$. Let us denote by $\mathcal{U}$ the set of admissible control functions $(x, t) \mapsto u(x, t)$ and by $\mathcal{F}$ the space of functions $(t, x) \mapsto \phi(x, t)$ where the problem is formulated. In the following, we properly define $\mathcal{U}$ and $\mathcal{F}$ for the optimal control problem

$$\inf_{u \in \mathcal{U},\, \phi \in \mathcal{F} :\, (8) \text{ holds}} J(u, \phi) \tag{9}$$

where $J : \mathcal{U} \times \mathcal{F} \to \mathbb{R}$ is a smooth cost functional.

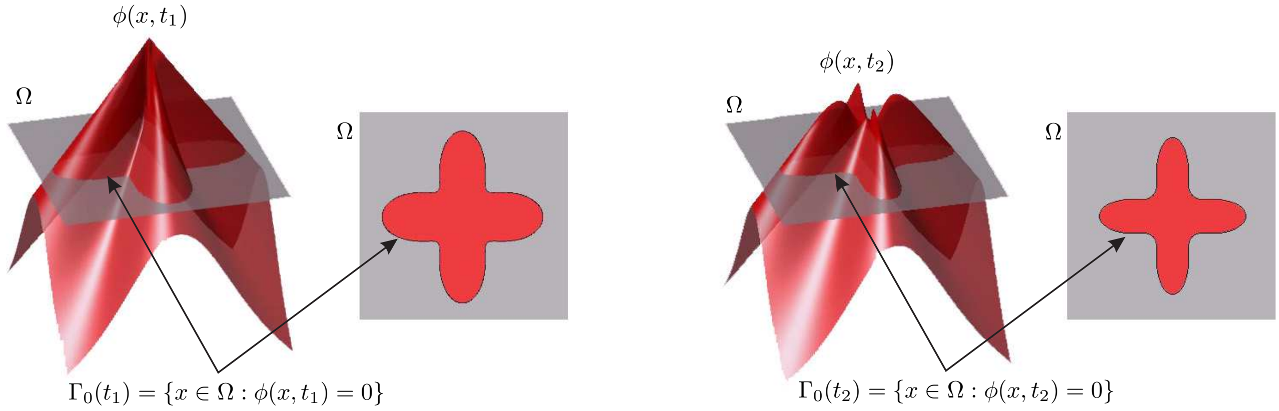
$\phi(x,t_1)$
Ω
Ω
$\Gamma_0(t_1) = \{x \in \Omega : \phi(x,t_1) = 0\}$
$\phi(x,t_2)$
Ω
Ω
$\Gamma_0(t_2) = \{x \in \Omega : \phi(x,t_2) = 0\}$

Figure 3: Fronts described by the zero LSs of a multidimensional function ϕ at two different time instants t_1 and t_2.

Let $A \subset \mathbb{R}^n$ be open and define

$$[u]_1 := \sup_{x,\,y \in A,\, x \neq y} \frac{|u(x) - u(y)|}{|x - y|}$$

where $u : A \to \mathbb{R}$. We denote the class of continuous functions and bounded continuous functions in A by $C^0(A)$ and $C^0_b(A)$, respectively. Moreover, let

$$C^{0,1}(A) := \left\{ u \in C^0_b(A) : [u]_1 < \infty \right\} .$$

Then, $(C^{0,1}(A), \|\cdot\|_1)$ is complete, where $\|u\|_1 := \|u\|_\infty + [u]_1$, or, in other words, $C^{0,1}(A)$ endowed with the norm $\|\cdot\|_1$ is a Banach space (see, e.g., (Alessandri, Bagnerini, and Gaggero, 2013)). Note that, if $[u]_1 < \infty$, then $[u]_1$ is just the smallest constant L such that $|u(x) - u(y)| \leq L\,|x - y|$, $x, y \in A$, i.e., u is uniformly Lipschitz. Moreover, since a Lipschitz function is uniformly continuous and therefore continuously extendable to the boundary of its domain, it follows that

$$C^{0,1}(A) = C^{0,1}(\overline{A}) .$$

Based on the aforesaid, the following propositions hold (Fornaro, Maniglia, and Metafune, 2004).

Proposition 1 *If A is bounded, the immersion $C^{0,1}(A) \hookrightarrow C^0(\overline{A})$ is compact, i.e., if $(u_n)_{n\in\mathbb{N}}$ is a sequence of functions in $C^{0,1}(A)$, there exists a subsequence $(u_{n_k})_{k\in\mathbb{N}}$ of $(u_n)_{n\in\mathbb{N}}$ that converges uniformly in $\overline{A}$.*

Proposition 2 *Let $(u_n)_{n\in\mathbb{N}} \in C^{0,1}(A)$ be a sequence that converges uniformly to u and such that $[u_n]_1 < c$ for some $c > 0$. Then $u \in C^{0,1}(A)$ and $[u]_1 < c$.*

Let us now consider (8), which is rewritten in the form of the more general Hamilton-Jacobi equation

$$\phi_t(x,t) + H(x,t,\nabla\phi(x,t)) = 0 \text{ in } \Omega \times (0,T) \tag{10}$$

where $H(x,t,p) = u(x,t)\,|p|$ is the Hamiltonian function. Consider also initial conditions $\phi(x,0) = \phi_0(x)$ in $\overline{\Omega}$. In the following, we will focus on viscosity solutions "inside Ω" and viscosity supersolutions in $\partial\Omega$ (Capuzzo-Dolcetta and P.L. Lions, 1990).

We need to assume the following.

Assumption 1 *Let $u \in C^0(\overline{\Omega} \times [0,T])$ such that $u(x,t) > 0$ for $x \in N(\partial\Omega)$ and $t \in [0,T]$.*[1]

Assumption 2 *There exists $L > 0$ such that $|u(x,t) - u(y,t)| \leq L\,|x-y|$ for all $x, y \in \Omega$ and $t \in [0,T]$.*[2]

Assumption 3 *There exists $M > 0$ such that $|u(x,t_1) - u(x,t_2)| \leq M\,|t_1 - t_2|$ for all $x \in \overline{\Omega}$ and $t_1, t_2 \in [0,T]$.*[3]

Theorem 1 *If $\phi_0 \in C^0(\overline{\Omega})$, there exists a viscosity solution $\phi \in C^0(\overline{\Omega} \times [0,T])$ for (10) such that $\phi(x,0) = \phi_0(x)$ in $\overline{\Omega}$.*

Proof. See (Capuzzo-Dolcetta and P.L. Lions, 1990, Theorem IV.2, p. 655) with all the required assumptions satisfied since more restrictive conditions hold because of the specific choice of the Hamiltonian function. Moreover, such a result states that ϕ is a viscosity supersolution on $\overline{\Omega} \times (0,T)$. □

It is worth noting that the viscosity solution $\phi \in C^0(\overline{\Omega} \times [0,T])$ is not unique in general. However, it is the minimum viscosity supersolution $v(x,t)$ of (10) on $\overline{\Omega} \times [0,T]$ such that $v(x,0) \geq \phi_0(x)$ on $\overline{\Omega}$. Moreover, such a solution is Lipschitz w.r.t. x near $\partial\Omega$, as it follows from the proof of (Capuzzo-Dolcetta and P.L. Lions, 1990, Theorem IV.2, p. 655). In fact, Assumption 1 holds and $H(x,t,p) \to +\infty$ as $|p| \to +\infty$ uniformly for $x \in N(\partial\Omega)$ and $t \in [0,T]$.

Let

$$\mathcal{F} := \left\{ \phi \in C^0(\overline{\Omega} \times [0,T]) : \phi \text{ is Lipschitz on } N(\partial\Omega) \text{and a solution of } (10) \right\}$$

and, for some $a > 0$,

$$\mathcal{U} := \left\{ u \in C^{0,1}(\overline{\Omega} \times [0,T]) \text{ such that } u(x,t) \geq a\,,\, x \in N(\partial\Omega), t \in [0,T] \right\}.$$

As it will be clearer from what follows, $\mathcal{U}$ is the set of the admissible controls for a given cost functional that satisfies the next assumption.

Assumption 4 *Let $J : \mathcal{U} \times \mathcal{F} \to [0,\infty)$ such that $J(\cdot, s) : \mathcal{U} \to [0,\infty)$ is lower semicontinuous for $s \in \mathbb{R}$.*

Therefore, let us recast the problem (9) as follows:

$$\inf_{u \in \mathcal{U},\, \phi \in \mathcal{F}} J(u,\phi)\,. \tag{11}$$

[1] The positivity assumption in a neighborhood of $\partial\Omega$ allows one to rely on the existence of supersolutions in $\overline{\Omega}$ (Capuzzo-Dolcetta and P.L. Lions, 1990, Proposition II.2, p. 647).

[2] In principle, we may adopt a more general assumption, i.e., $|u(x,t) - u(y,t)| \leq \omega(|x-y|)$, where $\omega : [0,\infty) \to [0,\infty)$ is a continuous, nondecreasing, and subadditive function such that $\omega(0) = 0$ (see (Capuzzo-Dolcetta and P.L. Lions, 1990, Assumption (H2), p. 648)).

[3] Likewise in Assumption 2, we may relax such assumption (see (Capuzzo-Dolcetta and P.L. Lions, 1990, eq. (36), p. 655)).

Theorem 2 *There exists $u^* \in \mathcal{U}$ such that*

$$J(u^*, \phi^*) = \inf_{u \in \mathcal{U},\, \phi \in \mathcal{F}} J(u, \phi)$$

for some $\phi^* \in \mathcal{F}$.

Proof. Since J is lower bounded, there exists a minimizing sequence $(u_k)_{k \in \mathbb{N}}$ for J in $\mathcal{U}_a$. For every $k \in \mathbb{N}$, let Φ_k be a viscosity solution of

$$\frac{\partial}{\partial t}\Phi_k(x,t) + u_k(x,t)\,|\nabla \Phi_k(x,t)| = 0 \text{ in } \Omega \times (0,T) \tag{12}$$

such that $\Phi_k(x,0) = \phi_0(x)$, where $\Phi_k \in C^0(\overline{\Omega} \times [0,T])$ is Lipschitz on $N(\partial\Omega)$. Since u_k belongs to $\mathcal{U}$, from Propositions 1 and 2 it follows that there exists a subsequence of $(u_k)_{k \in \mathbb{N}}$ that uniformly converges in $\overline{\Omega} \times [0,T]$ to some $u^* \in C^{0,1}(\overline{\Omega} \times [0,T])$. To reduce the notational overhead and with a little abuse of notation, we will denote such a subsequence with u_k and so it will be also for other sequences.

First of all, let us verify that u^* belongs to $\mathcal{U}$. Toward this end, notice that $u^*(x,t) \geq a$ for $x \in N(\partial\Omega)$ and $t \in [0,T]$. Since for every $\varepsilon > 0$ there exists $k_\varepsilon \in \mathbb{N}$ such that, for $k > k_\varepsilon$, it follows that $u^*(x,t) > u_k(x,t) - \varepsilon$ for $(x,t) \in \overline{\Omega} \times [0,T]$, if $x \in N(\partial\Omega)$ we obtain $u^*(x,t) > a - \varepsilon$. From the arbitrariness of ε it follows $u^*(x,t) \geq a$ for $x \in N(\partial\Omega)$ and $t \in [0,T]$ (from now on, we omit to recall such a dependence on t for the sake of brevity).

Then, let us focus on (12), where u_k is the subsequence converging to u^* we considered before; let us show that $\Phi_k \in C^0(\overline{\Omega} \times [0,T])$ converges to some $\phi^* \in C^0(\overline{\Omega} \times [0,T])$ such that

$$\frac{\partial}{\partial t}\phi^*(x,t) + u^*(x,t)\,|\nabla \phi^*(x,t)| = 0 \text{ in } \Omega \times (0,T) \tag{13}$$

with $\phi^*(x,0) = \phi_0(x)$ by passing to a subsequence if necessary. Owing to the structure of the Hamiltonian $H(x,t,p)$ in (10) (it is convex in p and such that $H(x,t,p)$ tends to $+\infty$ as $|p| \to +\infty$ uniformly for $x \in N(\partial\Omega)$), there exist $\alpha > 0$ and $\beta > 0$ such that $H(x,t,p) \geq \alpha|p| - \beta$. From such inequality and Assumption 3, it follows that $\nabla\Phi_k$ is uniformly bounded w.r.t. k in some $N(\partial\Omega)$. Thus, using (13) we get that also Φ_k is uniformly bounded, and so there exists $\delta > 0$ such that $\|\Phi_k\|_1 \leq \delta$ in some $N(\partial\Omega)$. Using the arguments in (M.G. Crandall and P.L. Lions, 1985, Theorem 1, p. 385), it follows that an a-priori estimate of the modulus of continuity of Φ_k near $\partial\Omega$ propagates in Ω. Hence, Φ_k is bounded in $C^{0,1}(\overline{\Omega} \times [0,T])$ and, thanks to Propositions 1 and 2, it admits a subsequence that uniformly converges to some ϕ^*, which is Lipschitz in some $N(\partial\Omega)$. By passing to a further subsequence if necessary, for u_k and Φ_k, using (M.G. Crandall and P.L. Lions, 1986, Theorem 1.4, p. 375), we obtain (13).

If Φ_k is a viscosity supersolution in $\overline{\Omega} \times (0,T)$ and it is Lipschitz in $N(\partial\Omega)$, then u_k is Lipschitz in $\overline{\Omega}$, Φ_k and u_k admit subsequences that uniformly converge to ϕ^* and u^*, ϕ^* is a viscosity supersolution in $\overline{\Omega} \times (0,T)$ of (13). Therefore, we finally obtain that ϕ^* belongs to $\mathcal{F}$ and, using (Kurdila and Zabarankin, 2005, Proposition 7.1.2, p. 206) with Assumption 4, we get

$$J(u^*, \phi^*) \leq \liminf_{k \to +\infty} J(u_k, \Phi_k)$$

which concludes the proof.

□

2.3 EKF-based optimization

According to (Alessandri et al., 2014; Alessandri, Bagnerini, Gaggero, and Traverso, 2016), we rely on the ERIM to find approximations of the solution of (5) by replacing the control law with a parametrized one and directly tuning its parameters to optimize the considered performance index. For instance, a typical choice relies on constraining the control law to take on the form of a one-hidden-layer feedforward neural network, whose weights have to be properly selected. The goal consists in approximating the mapping $(x,t) \mapsto u^\circ(x,t)$ that solves the problem (5) by linear combinations of parametrized basis functions (Barron, 1993; Kůrková and Sanguineti, 2002). According to the ERIM paradigm, we impose

$$u(x,t) = \gamma(x,t,w) \tag{14}$$

in the normal flow equation (3), where γ is the aforementioned parametrized control law and $w \in \mathbb{R}^m$ is a vector of m parameters to be tuned. It is worth noting that, using (14), both ϕ and u become functions of w, and therefore also the cost functional J turns out to depend on w. Thus, problem (5) reduces to find the optimal weights w^o that minimize the cost J, i.e.,

$$w^\mathrm{o} \in \underset{w \in \mathbb{R}^m \text{ s.t. (3) and (14) hold}}{\operatorname{argmin}} J(w). \tag{15}$$

Problem (15) is a mathematical programming one that can be solved by gradient-based methods. Toward this end, it is necessary to compute the gradient of J with respect to the parameters w with (3) and (14) as constraints. In general, such a step is difficult due to the nonlinear dependence of both J and γ on the parameters w. However, as proposed by Alessandri et al. (2014,1), it is possible to find an analytic expression for the gradient of J by solving an adjoint equation. In the case of the normal flow (3), the adjoint equation to be solved backward in time is the following (see (Alessandri et al., 2016) for the details):

$$\begin{cases} -\mu_t = (\mu\gamma F_1)_x + (\mu\gamma F_2)_y - h_\phi(\phi,\gamma) \text{ in } \Omega \times [0,T) \\ \mu(x,T) = -\bar{h}_\phi(\phi(x,T)) \text{ in } \Omega\,, \end{cases} \tag{16}$$

where $\mu(x,t)$ is the Lagrange multiplier. Moreover,

$$F_1 := \frac{\phi_x}{|\nabla\phi|} \qquad F_2 := \frac{\phi_y}{|\nabla\phi|}\,.$$

By solving (16), we can obtain the expression for the gradient of the cost with respect to w:

$$\nabla_w J(\phi,u) = \int_0^T \int_\Omega \Big(h_u(\phi,\gamma) + \mu\,|\nabla\phi| \Big) \nabla_w\gamma \, dx\, dt\,. \tag{17}$$

When performing the search for the optimal weights, the initial guess of w may give rise to different final results due to local minima that can affect (15), and therefore convergence to a global optimum may be undermined. To overcome this difficulty, multistart techniques may be adopted. They consists in repeating the optimization with different, randomly-chosen initial weights, and selecting as the optimal ones those providing the smallest cost. Such an approach for the solution of (15), referred to as "multistart optimization" in the following, is described in Algorithm 1, where the number of initial guesses for the weights is denoted by L.

In the next section, we will present an alternative optimization approach based on the EKF to mitigate the issue of local minima trapping, as it will be shown by means of simulations.

In learning from data it is often necessary to perform regression on input/output pairs $\{(z_k, y_k)\,,\ k = 0,1,\ldots\}$, where $z_k \in Z \subset \mathbb{R}^p$ and $y_k \in Y \subset \mathbb{R}$ by using a parametrized function ζ to approximate the mapping $z_k \mapsto y_k$, i.e., $\tilde{y}_k = \zeta(z_k, w)$, where $\tilde{y}_k \in Y$ is the approximation of y_k, $w \in \mathbb{R}^m$ is a

Algorithm 1 for the selection of the optimal weights

```
procedure MULTISTART OPTIMIZATION
Inputs:
    equations (3), (16), (17), and $L$
Outputs:
    vector of the optimal parameters $w^\circ$
    optimal cost $J^\circ$
Main loop:
    $J^\circ \leftarrow +\infty$
    for $l$ from 1 to $L$ do
        generate a random initial choice of $w_0(l)$
        $k \leftarrow 0$
        while (stopping criterion is not satisfied) do
            $u(x,t) \leftarrow \gamma(x,t,w_k(l))$
            solve the normal flow equation (3)
            solve the adjoint equation (16)
            compute the gradient (17)
            $w_{k+1}(l) \leftarrow$ descent step starting from $w_k(l)$
            $k \leftarrow k+1$
        end while
        $w^\circ(l) \leftarrow w_k(l)$
        $J^\circ(l) \leftarrow J(w_k(l))$
        if $(J^\circ(l) < J^\circ)$ then
            $J^\circ \leftarrow J^\circ(l)$
            $w^\circ \leftarrow w^\circ(l)$
        end if
    end for
end procedure
```

vector of parameters to be properly tuned, and Z and Y are compact sets (Ilin, Kozma, and Werbos, 2008).

In the following, we will focus on parametrized functions ζ that are of the same family of the control law γ used in (14) to attack the optimal control of level sets with the ERIM. In practice, the problem of finding the best value of the coefficients w may be recursively solved as a neural network training one using the EKF, which can be regarded as the efficient application of the Gauss-Newton method (Bertsekas, 1996). In this section, we motivate the use of this approach and adapt it to our problem.

Following a large literature (see, e.g., (Alessandri, Cuneo, Pagnan, and Sanguineti, 2007; Iiguni, Sakai, and Tokumaru, 1992; Mussa and Glen, 2010; Nishiyama and Suzuki, 2001; Schottky and Saad, 1999)), the learning from data based on the EKF can be stated as a nonlinear filtering problem with system equations given by

$$w_{k+1} = w_k \tag{18a}$$

$$y_k = \zeta(z_k, w_k) + v_k, \tag{18b}$$

where $k = 0, 1, \ldots$. It is implicitly assumed that the input/output pairs are generated by some optimal vector of weights $w^* \in \mathbb{R}^m$, i.e., $y_k = \zeta(z_k, w^*)$. Such optimal weights are unknown, and they are estimated by using the simple state-augmented dynamics (18a), whereas (18b) represents an output equation. The scalar quantity v_k is regarded as a measurement noise, i.e., it is a random variable that accounts for the difference between y_k and $\zeta(z_k, w_k)$. The estimate $\hat{w}_k \in \mathbb{R}^m$ of w^* can be computed by using the standard equations of the EKF, i.e.,

$$\hat{w}_{k+1} = \hat{w}_k + K_k \left(y_k - \zeta\left(z_k, \hat{w}_k\right) \right) \;\; k = 0, 1, \dots \tag{19a}$$

with

$$K_k := P_k H_k \left(H_k^\top P_k H_k + r_k \right)^{-1} \tag{19b}$$

$$P_{k+1} = P_k - K_k H_k^\top P_k \,, \tag{19c}$$

where $P_k \in \mathbb{R}^{m \times m}$ is a symmetric, positive definite matrix, $K_k \in \mathbb{R}^m$, $r_k \in \mathbb{R}$ denotes the covariance of the measurement noise, and

$$H_k := \nabla_w \zeta(z_k, \hat{w}_k) \,. \tag{19d}$$

Of course, we have to initialize (19) with some guess of the initial weights $\hat{w}_0$ and a symmetric, positive definite matrix P_0. It is known (see (Reif, Günter, Yaz, and Unbehauen, 1999)) that the stability properties of the estimation error can be ensured only under the assumption of zero-mean, random noises with finite covariance.

The application of (19) to the problem of level set control is not straightforward. In fact, in (19) we assume to know the values of the output y_k. In our context, the role of the output is played by the optimal cost, which is unknown. The values of the cost corresponding to w_k are the equivalent of the mapping ζ. Thus, instead of (18b) we have

$$J(w^\circ) = J(w_k) + v_k,$$

where $J(w^\circ)$ is unknown. In addition, since $J(w^\circ) \leq J(w_k)$ for all $w_k \in \mathbb{R}^m$, the noise v_k results always to be nonpositive instead of having a mean equal to zero as previously discussed. This motivates the introduction of a bias $b_k = J(w^\circ)$, which allows to model the lack of knowledge on $J(w^\circ)$ and it is estimated together with the optimal weights. Thus, instead of (18) we rely on the following model:

$$w_{k+1} = w_k \tag{20a}$$

$$b_{k+1} = b_k \tag{20b}$$

$$0 = J(w_k) - b_k + v_k, \tag{20c}$$

where $k = 0, 1, \dots$. The computation of the gain K_k to apply the EKF is, *mutatis mutandis*, likewise in (19). In other words, we perform an estimate of the optimal weights given by (19a) with a system state (w_k, b_k) and gain K_k computed as in (19b) and (19c) with $P_k \in \mathbb{R}^{(m+1)\times(m+1)}$, $K_k \in \mathbb{R}^{m+1}$, $y_k = 0$, $J(w_k) - b_k$ instead of $\zeta\left(z_k, \hat{w}_k\right)$ in (19a), and

$$H_k := \begin{pmatrix} \nabla_w J(\hat{w}_k) \\ -1 \end{pmatrix} .$$

In this way, the quantity $-b_k + v_k$ has zero mean, and all the properties of the EKF hold. Since the exact value of r_k is unknown, we rely on the adaptive estimate proposed by Iiguni et al. (1992)[equation (36), p. 962].

It is worth noting that classical results on bias estimation with the Kalman filter concerning the reduction of computational effort (see (Caglayan and Lancraft, 1983; Keller and Darouach, 1999)) are not useful here, as the bias b_k is just scalar as compared with the much larger dimension of w_k. By contrast, the comparison with the multistart method presented in Algorithm 1 as to both computational demand and effectiveness of the optimization is fundamental.

Algorithm 2 for the selection of the optimal weights

```
procedure EKF-BASED OPTIMIZATION
Inputs:
    equations (3), (16), (17), (19), and (20)
Outputs:
    vector of the optimal parameters $w^\circ$
    optimal cost $J^\circ$
Main loop:
    generate a random initial choice of $w_0$
    $k \leftarrow 0$
    while (stopping criterion is not satisfied) do
        $u(x,t) \leftarrow \gamma(x,t,w_k)$
        solve the normal flow equation (3)
        solve the adjoint equation (16)
        compute the gradient (17)
        $w_{k+1} \leftarrow$ EKF step (19) starting from $w_k$
        $k \leftarrow k+1$
    end while
    $w^\circ \leftarrow w_k$
    $J^\circ \leftarrow J(w_k)$
end procedure
```

The procedure to solve (15) by means of the EKF-based optimization is detailed in Algorithm 2. Clearly, its main advantage as compared with Algorithm 1 is the computational saving, as it is not required to repeat the random selection of the initial weights and run the optimization for each of them. In fact, we will show by means of simulations in the next section that the optimization based on the EKF is very robust to local minima trapping, and therefore only one guess for the initial weights is enough to guarantee satisfactory results.

2.4 Simulation results

The effectiveness of the EKF-based optimization in comparison with the multistart optimization in a complex tracking example involving a change of topology has been investigated by means of an extensive simulation campaign. We have addressed the problem to find $u(x,t)$ such that the zero level set of the function $\phi(x,t)$ tracks a reference front $\phi_{\text{ref}}(x,t)$. We have considered the following cost measuring the difference between the reference and actual level sets:

$$J = \int_0^T \int_\Omega \Big(\hat{H}(\phi(x,t)) - \hat{H}(\phi_{\text{ref}}(x,t)) \Big)^2 \, dx \, dt, \tag{21}$$

where $\hat{H}$ is a smooth approximation of the Heaviside step function ($H(z) = 1$ if $z \geq 0$ and $H(z) = 0$ otherwise) proposed by Yang and Tomlin (2013) and given by

$$\hat{H}(z) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{z}{\tau}\right)$$

for a small value of τ (we fixed τ to 10^{-2}).

The final time instant T has been chosen equal to 1.5 with a sampling time $\Delta t = 0.03$, which corresponds to a total number of 50 time steps. The spatial domain Ω has been chosen as $[-0.75, +0.75] \times [-0.5, +0.5]$. The normal flow equation (3) and the corresponding adjoint equation (16) have been

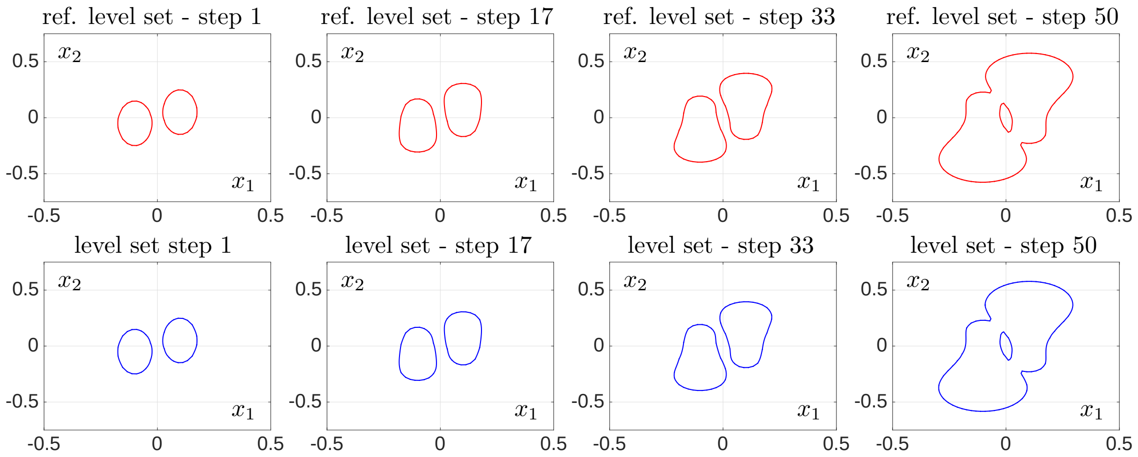

Figure 4: Front tracking snapshots obtained with the EKF-based optimization.

solved numerically on a regular Cartesian grid composed of 75×50 nodes using a second-order numerical finite differences scheme both in space and time. Particular attention has been paid to the discretization of the adjoint equation, especially as regards the boundary conditions. Since both equations are of hyperbolic type, upwind approximations are used to avoid numerical instabilities. For this purpose, we have performed spatial discretization by using an upwind second-order essentially non-oscillatory (ENO) scheme (Kimmel, 2004, chap. 3). The time discretization is treated by a second-order total variation diminishing Runge-Kutta scheme. The level set methods are handled by using the Matlab toolbox by Mitchell (2008).

All the simulations have been performed on a personal computer equipped with a 2.6 GHz Intel Xeon CPU and 64 GB of RAM. The multistart optimization relies on the routine *fmincon* of the Matlab Optimization Toolbox by using the sequential quadratic programming algorithm and the analytical expression (17) of the gradient of the cost. As regards the EKF-based optimization, we have fixed the covariance matrix P_0 of the initial weights equal to the identity matrix of size $(m+1)\times(m+1)$ and the initial covariance of the measurement noise equal to 0.01. Such values have been found after an experimental tuning with the aim of finding the ones providing the fastest convergence to the optimal cost.

As regards the parametrized structure for the control law, we have focused on one-hidden-layer feedforward neural networks with sigmoidal activation functions with 5 neurons. The reference curve has been obtained with (3) by assigning $u(x,t)$. In particular, we have adopted the following speed for all $x \in \Omega$ and $t \in [0,T]$:

$$u(x,t) = \frac{1}{4}|\sin(\pi t)|x_1^2 + 6x_2^2. \tag{22}$$

In order to obtain a value for the true optimal weights w^*, which will be useful to evaluate the performances of the considered approaches, we have approximated the mapping (22) using the Levenberg-Marquardt algorithm available in Matlab (function *trainlm*) with 5000 input/output pairs, and we have extracted the weights from the resulting trained network.

For the sake of comparison, we have performed the optimization using both Algorithm 1 for the multistart method and Algorithm 2 for the EKF learning starting from a total of $L = 1000$ different initial values for the weights. Let us denote by $w^\circ(l)$, $l = 1, \dots, L$, the weights at the end of the optimization procedure with the l-th value of the initial weights. Then, we consider the distance from the true optimal ones, i.e., $\tilde{w}(l) := |w^\circ(l) - w^*|$, $l = 1, \dots, L$. Furthermore, let $J^\circ(l)$ and $\overline{T}(l)$ be the optimal cost and the time needed to complete the optimization with the l-th value of the initial weights for $l = 1, \dots, L$, respectively.

The results of the tracking obtained with the EKF-based optimization are shown in Fig. 4. It turns out that the proposed approach has nice tracking capabilities. In fact, the reference curve is

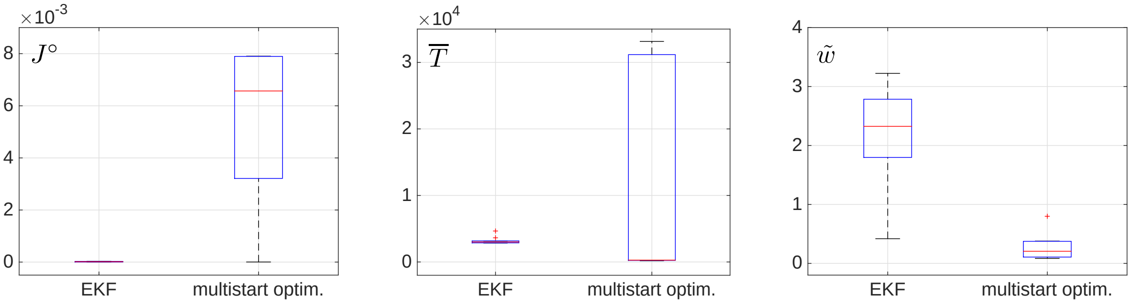

Figure 5: Boxplots of optimal cost, mean computational time, and distance from the optimal weights.

tracked with good accuracy, and also the change of topology occurring at the end of the simulation horizon is correctly dealt with.

Fig. 5 shows the boxplots of the optimal cost J°, of the time $\overline{T}$ needed to complete each optimization, and of the distance $\tilde{w}$ computed over the considered 1000 different initial weights for both the multistart optimization and the EKF-based one. Looking at the obtained results, we can argue that both methods are reliable since the minimum values of the optimal costs are comparable. However, the optimal costs of the multistart optimization are characterized by a large dispersion around the median, whereas the costs of the EKF-based approach are all concentrated around the median. This is due to the local minima trapping issue that affects the optimization with descent methods such as the sequential quadratic programming. A large number of initial weights is required to obtain a value of the weights providing satisfactory tracking capabilities. By contrast, few initial guesses are enough if the EKF is used, as it is able to successfully escape from local minima most of the times.

Concerning the computational overhead, the dispersion around the median of the optimization time $\overline{T}$ for the multistart optimization is experienced to be quite large: the median is lower than the corresponding one of the EKF since the stopping criteria of the optimization routine of the former are rapidly satisfied when a local minimum is found. The property of local minima escaping of the EKF is confirmed by the fact that the same time is needed to complete the optimization for almost all the considered initial weights, as the stopping criteria are not satisfied after few iterations due to the trapping into a local minimum.

A special discussion concerns the measure of the distance of the weights from the true optimal ones, which si given by $\tilde{w}$. In this case, on the average the EKF-based optimization provides weights that are far from the true optimal ones. However, all such weights correspond to low and similar values of the cost. This behavior may be regarded as a confirmation of the presence of many global minima in (15) that are far one from the other but all guarantee almost the same values of the cost (Fukumizu and Amari, 2000; Sussmann, 1992).

To sum up, the EKF-based optimization turns out to be preferable with respect to the multistart one in this example. In fact, its robustness to local minima trapping allows one to obtain savings in the overall computational effort, as in general it is sufficient to start from a unique guess for the initial weights to find "good" optimal ones. By contrast, the best number of trials in the multistart optimization is not known *a priori*, as it may range from few ones in the most lucky cases to a large number in the worst ones, thus making such an approach less appealing in practice.

3 Laboratory setup for Identification and Control of Level Sets Models

We have constructed a test rig allows to verify the effective capability to control the interface between two fluids. Toward this end, we need to account for the physics of the process and develop a suitable description of the interface. Unfortunately, the resulting model based on the cascade of the physical setup based on Navier-Stokes equations and level sets turns out to be very computationally demanding if one wants to use it for the purpose of optimal control during online operations.

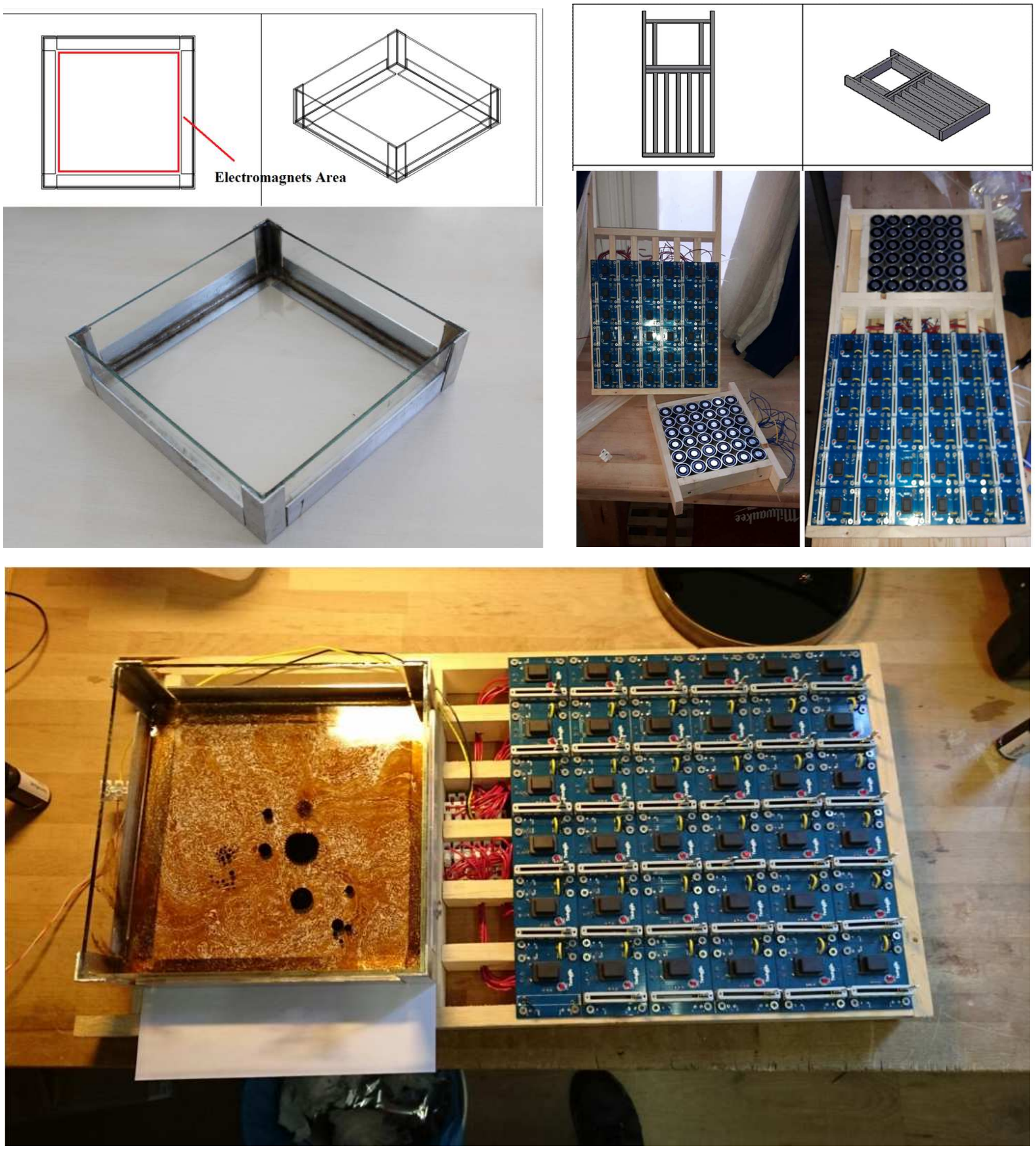

Figure 6: Glass vessel, electronic package, and the complete setup.

Thus, we have identified a black-box model of the cascade of Navier-Stokes and level set equations by using nonlinear approximators such as neural networks. The goal is the fast computation of the optimal control inputs in real time, without requiring the online numerical solution of a cascade of two PDEs.

3.1 Modeling the experimental setup

The test rig is composed of a tank of water and ferrofluid under the effect of a matrix of driving electromagnets. These electromagnets can change the magnetic field in order to modify the shape of the ferrofluid. A ferrofluid is a liquid in which ferromagnetic nanoparticles are suspended in a carrier liquid. Specifically, we focus on the case of a light polar mineral oil. The size of the particles prevents their attraction since the inter-magnetic forces are low with respect to the surfactant's Van der Waals ones. The ferrofluid is mixed with a salt-saturated water solution that provides total separation between the two fluids. The overall density is almost equal to the density of the single components to avoid strong stratification. A wooden frame is used to keep the setup together with a glass vessel to house the fluid and the electronic package to drive the magnets. A picture of the used testbed is reported in Fig. 6.

The model is based on the following assumptions: (i) both fluids are in the liquid state, (ii) both fluids are incompressible, and (iii) both fluids are Newtonian, i.e., their viscosity is independent from the flow speed, and therefore it is possible to consider their viscosity constant if we assume a constant temperature. Under these hypotheses, the fluid dynamics can be represented by the following dimensionless incompressible 2D Navier-Stokes equations:

$$u_t + p_x = -uu_x - vu_y + \frac{1}{Re}(u_{xx} + u_{yy}) + \frac{1}{Fr^2}\, g_X \tag{23a}$$

$$v_t + p_y = -uv_x - vv_y + \frac{1}{Re}(v_{xx} + v_{yy}) + \frac{1}{Fr^2}\, g_Y \tag{23b}$$

$$u_x + v_y = 0 \tag{23c}$$

where u and v are the x and y velocity components, respectively, p is the pressure, Re is the Reynolds number, Fr is the Froude number, and g_X and g_Y are the x and y acceleration field components, respectively.

The momentum equations (23a) and (23b) describe the time evolution of the velocity field (u, v). The incompressibility condition (23c) is not a time-dependent equation, but an algebraic condition. If we consider p as a Lagrange multiplier, it is possible to obtain a new form of the Navier-Stokes equations that is easier to solve numerically (Seibold, 2008b). Thus, we focus on the following equations instead of (23):

$$u_t + p_x = -(u^2)_x - (uv)_y + \frac{1}{Re}(u_{xx} + u_{yy}) + \frac{1}{Fr^2} g_X \tag{24a}$$

$$v_t + p_y = -(v^2)_y - (uv)_x + \frac{1}{Re}(v_{xx} + v_{yy}) + \frac{1}{Fr^2} g_Y\,. \tag{24b}$$

The acceleration field g is the resultant vector of the acceleration field g_m caused by the magnetic force acting on the ferrofluid and of the acceleration field g_t caused by the interfacial tension. The acceleration g_m depends on the magnetic field B generated by 36 electromagnets that are arranged in a 6×6 grid. To compute the magnetic field generated by each electromagnet, we use the equations proposed by Callaghan and Maslen (1960) and Derby and Olbert (2010). Then, the overall field can be obtained from the field generated by each magnet using the classical superposition principle. Specifically, let us collect in the vector a the current intensity of the various electromagnets. Thus, the overall magnetic field B is a function of the vector a as follows:

$$B = h(a) \tag{25}$$

where, for details on the function h, we refer to (Callaghan and Maslen, 1960; Derby and Olbert, 2010).

To compute the force acting on each magnetic particle of the ferrofluid, we use the following equation (Scherer and Figueiredo Neto, 2005):

$$F_m = \nabla B \cdot \mu \tag{26}$$

where μ is the magnetic moment of each particle. From F_m it is possible to obtain an approximate value of g_m. In fact, consider a small element of ferrofluid with volume V_e. Since the ferrofluid behaves like a homogeneous fluid (Petit, Kedous-Lebouc, Avenas, Tawk, and Artega, 2011), the force F acting on the entire element of ferrofluid is given by

$$F_{m,e} = n_p F_m \tag{27}$$

where n_p is the number of magnetic particles in the element. The acceleration g_m is then computed as

$$g_m = \frac{F_{m,e}}{\rho_f V_e} = \frac{n_p F_m}{\rho_f V_e} = \frac{n_p \mu}{\rho_f V_e} \nabla B \tag{28}$$

where ρ_f is the ferrofluid density.

The interfacial tension, caused by unbalanced attractive forces (Ghosh, 2009), acts on the particles of water and ferrofluid near the interface. Specifically, the acceleration field g_t caused by the interfacial tension can be expressed as follows:

$$g_t = \frac{1}{\rho} \sigma \kappa \, \delta(d) \, n \tag{29}$$

where ρ is the fluid density, σ is the interfacial tension coefficient, κ is the curvature of the interface, $\delta(d)$ is the Dirac δ function of the distance from the interface, and n is the normal versor to the interface (Sethian, 1999).

The evolution of the interface between water and ferrofluid is taken into account through the paradigm of level set methods. Thus, the interface is implicitly represented as the zero level set of a multidimensional function $\phi(x(t), t)$, where $x(t)$ is the position and t is the time. The evolution of ϕ is determined by the following Hamilton-Jacobi equation (Mitchell, 2007):

$$\phi_t + \nabla \phi(x(t), t) \cdot x'(t) = 0 \tag{30}$$

where the velocity field $x'(t)$ in (30) depends on the fluid dynamics of the system. Since the interface is the zero level set of ϕ, the normal n can be obtained as

$$n = \frac{\phi}{|\nabla \phi|} \, . \tag{31}$$

Using the expression of the normal, we can compute the curvature κ in (29) as

$$\kappa = \nabla \cdot n = \nabla \cdot \frac{\phi}{|\nabla \phi|} \tag{32}$$

whereas the distance from the interface d is given by

$$d = \left| \frac{\phi}{|\nabla \phi|} \right| . \tag{33}$$

Combining (32), (33), and (31), we get the expression for g_t as follows:

$$g_t = \frac{1}{\rho} \sigma \kappa(\phi) \delta \left(\frac{\phi}{|\nabla \phi|} \right) \frac{\phi}{|\nabla \phi|} \, . \tag{34}$$

In the following, we will address the problem of computationally tractable, approximate description of the model described so far.

3.2 Offline model identification

The dynamic model presented in the previous section allows one to describe the evolution in space and time of the interface between the water and the ferrofluid depending on the current intensity of the different electromagnets. However, it is very computational demanding, as it requires the numerical solution of two PDEs, i.e., the Navier-Stokes one and the level set Hamilton-Jacobi one (30). Such a computational difficulty may be a severe issue if the model is used to generate optimal control actions. Indeed, in this case it has to be executed several times to evaluate the effectiveness of a given control input over the others. Clearly, this can undermine the feasibility of the computation of optimal control actions on line.

Motivated by the desire of reducing the required computational effort, in this section we propose a black-box approach based on neural networks to approximate the functional relationship between the vector a and the evolution in time and space of the water-ferrofluid interface. More specifically, it is possible to write the following:

$$\phi = f(a, t) \tag{35}$$

where f is a function that results from the application of the model (24)-(34). Unfortunately, the analytic expression of f is unknown.

The idea is to approximate the unknown function f in (35) off line by using some nonlinear approximation techniques, in order to be able to generate the mapping $(a, t) \mapsto \phi$ almost instantaneously. In particular, the following procedure can be adopted off line:

- solve equations (24)-(34) for many different values of t and a and collect the corresponding pairs given by (a, t) and the function ϕ;
- apply some learning method to approximate such pairs.

More specifically, let us denote the different values of t and a as $t^{(i)}$ and $a^{(i)}$, respectively, for $i = 1, \ldots, N$, where N is the number of samples. Moreover, let $\phi^{(i)}$, $i = 1, \ldots, N$, be the corresponding function ϕ in the l.h.s. of (35).

In order to find an approximation of the function f in (35), we constrain it to take on a certain fixed structure given by

$$\phi = \gamma\left(a, t, w\right) \tag{36}$$

where γ is a parametrized function depending on the vector of parameters $w \in \mathbb{R}^p$. By tuning the values of such vector we can change the shape of the function γ. In our case, the goal is to search for the optimal parameters that yield a "good" interpolation of the pairs $(\tilde{a}^{(i)}, \phi^{(i)})$, $i = 1, \ldots, N$, where $\tilde{a}^{(i)} := (a^{(i)}, t^{(i)})$.

Among the various alternatives for the approximating function γ in (36), we focus on one-hidden-layer feedforward neural networks with sigmoidal activation functions. This choice is motivated by the availability of a huge literature on their approximating capabilities and the presence of efficient, ad-hoc developed algorithms software tools for the selection of the optimal parameters (Haykin, 2005). With this choice, in the case of scalar outputs the function γ has the following expression:

$$\gamma(\tilde{a}, w) = \sum_{i=1}^{\nu} c_i \, \sigma\left(\sum_{j=1}^{n} a_{ij}\tilde{a}_j + b_i\right) + c_0$$

where ν is the number of neurons, $\tilde{a}_j$ is the j-th component of $\tilde{a}$, σ is a sigmoidal activation function, c_0, c_i, $b_i \in \mathbb{R}$, and $a_i := \mathrm{col}(a_{i1}, \ldots, a_{in}) \in \mathbb{R}^n$, $i = 1, \ldots, \nu$. The vector of parameters to be optimized is given by $w = \mathrm{col}(a_1^\top, \ldots, a_\nu^\top, b_1, \ldots, b_\nu, c_0, \ldots, c_\nu)$. Their optimal value is usually found by solving the following optimization problem corresponding to a mean square error criterion:

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{N} \left|\phi^{(i)} - \gamma(\tilde{a}^{(i)}, w)\right|^2 \tag{37}$$

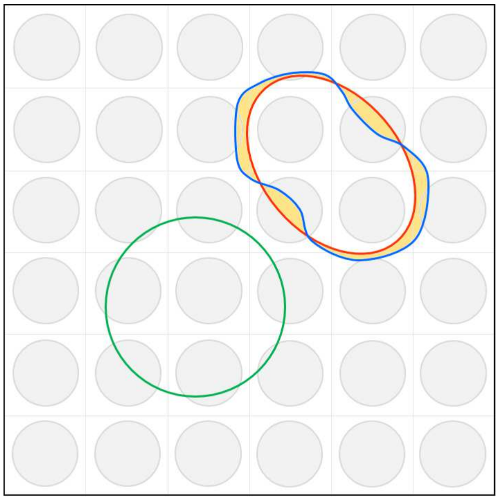

Figure 7: Initial condition of the simulations used to train the neural networks (green), example of ferrofluid shape (blue), and example of reference shape (red). The symmetric difference between the ferrofluid shape and the reference shape is given by the yellow area. The grey circles show the positions of the electromagnets.

Concerning the theoretical properties of this kind of neural networks, it is known that they are endowed with the universal approximation property, i.e., they are able to approximate with arbitrary accuracy any “well-behaved” function. Moreover, they are particularly well-suited to dealing with high-dimensional problems, as the number of parameters that are required to obtain satisfactory approximations grows only polynomially with the dimension of the inputs. The interested reader is referred to, e.g., (Barron, 1993; Zoppoli, Sanguineti, and Parisini, 2002) and the references therein for a deeper discussion.

The procedure to find the optimal values of the parameter vector w in (37) is called “training” in the neural network parlance, and a lot of efficient algorithms and corresponding software implementations exist in the literature, such as the classical backpropagation or Levenberg-Marquardt methods.

3.3 Optimal control of the ferrofluid interface

Given a reference shape, our purpose is to find a configuration of the electromagnets such that the ferrofluid is shaped as desired when the system has reached the steady state. In other words, we want to find the optimal values of the vector a containing the current intensity of the electromagnets so as to obtain desired shapes for the ferrofluid.

Toward this end, let us denote by Γ^d a reference shape for the ferrofluid, and let Γ be the actual shape obtained at regime as the zero level set of the function ϕ that is the output of the neural network as in (36). As measure of the distance between Γ and Γ^d, we adopt the symmetric difference, i.e., given two sets A and B, the symmetric difference is defined as $A \,\Delta\, B = (A \cup B) \setminus (A \cap B)$. Thus, we will use $\Gamma \,\Delta\, \Gamma^d$ as an index of performance (see Fig. 7). Summing up, we have to solve the following optimization problem:

$$a^\circ = \operatorname*{argmin}_{a \in \mathbb{R}^{36}} \Gamma \,\Delta\, \Gamma^d \tag{38}$$

At least in principle, problem (38) could be solved with any optimization routine. However, in this case the cost function does not change continuously with the input vector a. This is due to the spatial discretization used to solve equations (24)-(34). As a consequence, it is not possible to solve problem (38) using an optimization method that requires the computation of the gradient or higher-order derivatives of the cost function. Thus, the use of nonderivative methods, such as direct

Table 1: Results of the neural network training.

	mean(Δ_s)	mean(n_c)
$\nu = 5$	107.41	9.37
$\nu = 10$	93.78	6.17
$\nu = 15$	88.75	2.68
$\nu = 20$	86.51	2.02
$\nu = 25$	83.31	2.47
$\nu = 30$	82.93	1.38

search ones, is mandatory (Bertsekas, 1999). Specifically, we will adopt a generalized pattern search (GPS) algorithm.

To explain the principle of this algorithm, we show how it can be used to minimize a generic cost function $F(x)$ with $x \in \mathbb{R}^n$. Let us consider x_k at iteration k of the GPS algorithm, together with mesh size $\Delta x_k \in \mathbb{R}^+$. We evaluate $F(x)$ in the points $x_k^+ = x_k \pm \Delta x_k \cdot e_i$, $i \in \{1, ..., n\}$. The set of points x_k^+ is called pattern. The set of vectors e_i must be a spanning set of $\mathbb{R}^n$. We look for the x_k^+, such that $f(x_k^+) < f(x_k)$, which gives the best result (minimum value of $f(x_k^+)$). We denote such x_k^+ by x_k^o. If we find this x_k^o, we set $x_{k+1} = x_k^o$ and $\Delta x_{k+1} = 2\Delta x_k$. Otherwise, if no x_k^o is found, we set $x_{k+1} = x_k$ and $\Delta x_{k+1} = \Delta x_k/2$. This iteration continues until Δ_k is smaller than a certain tolerance value (Lewis and Torczon, 2000). Other stopping criteria may be adopted, such as the maximum number of performed iterations, the maximum number of cost function evaluations, and time limits.

3.4 Simulation results

First of all, we present the numerical results related to the construction of the black box model, and then we use it for the purpose of control.

To train the neural networks as described in Section 3.2, we need to collect the pairs $(\tilde{a}^{(i)}, \phi^{(i)})$ for $i = 1, \ldots, N$. Toward this end, we have solved 200 times the equations (24)-(34) starting from a certain initial condition and with different configurations of the electromagnets. Specifically, we have considered as spatial domain the rectangle $[-3, 3] \times [-3, 3]$. As initial condition we have assumed that the ferrofluid is in a circular shape, centered in (-1,-1) with radius 1.1, as shown in Fig. 7. Concerning the current intensity of the electromagnets, without loss of generality we have assumed that such intensity is restricted to take on three values, i.e., maximum one, half of the maximum, and zero.

The numerical solution of the Navier-Stokes equations has been obtained by suitably adapting the method proposed by Seibold (2008b) to take into account the characteristics of the model, i.e., the presence of two fluids with different physical properties and the effect of a magnetic force acting on the ferrofluid and interfacial tension. The spatial discretization has been done on a staggered grid, where u and v are placed on the vertical and horizontal cell sides, respectively, and p is in the cell center. The discretization step has been chosen equal to 0.1. Concerning the time discretization, we have selected a sampling time equal to 0.001. Furthermore, we have considered no-slip boundary condition on each wall of the tank for u and v and homogeneous Neumann boundary conditions for the pressure. As regards equation (30) modeling the evolution both in space and time of the interface between water and ferrofluid, again we rely on the Matlab toolbox on level set methods developed by Mitchell (2007).

The results of the 200 simulations described above have been saved at 50 different time steps. Thus, having at disposal a set of $N = 10000$ pairs $(\tilde{a}^{(i)}, \phi^{(i)})$, we have randomly divided such pairs into two subsets made up by 8700 and 1300 elements, corresponding to the training set (i.e., the pairs used to find the optimal values of the parameter vector w as in (37)) and the test set (i.e., the

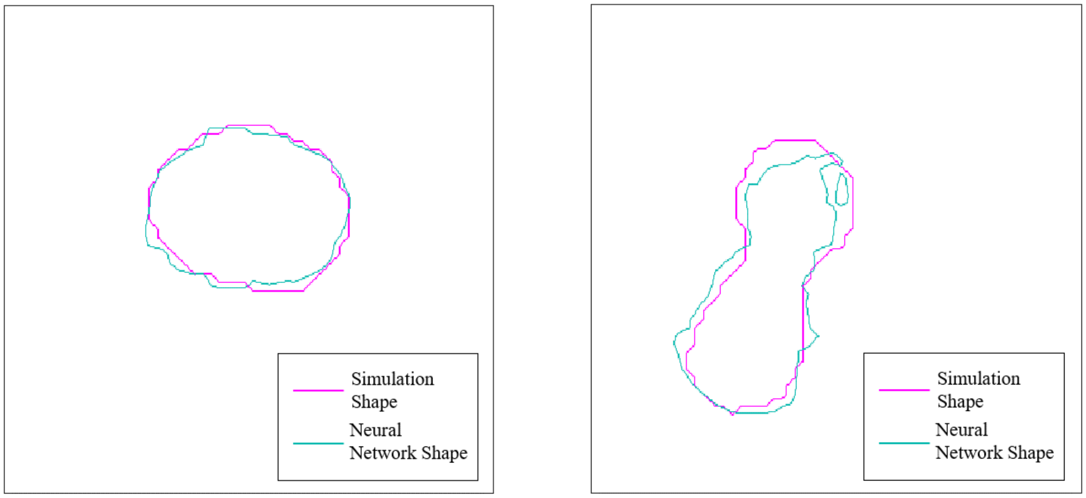

Figure 8: Examples of interfaces between water and ferrofluid obtained by solving equations (24)-(34) and by applying the neural network approximations.

pairs used to verify the effectiveness of the trained networks), respectively.

Different neural networks have been trained using the implementation of the Levenberg-Marquardt algorithm available in Matlab (function *trainlm*), corresponding to various numbers of neurons. Specifically, we have chosen ν in the range from 5 to 30.

The performances of the networks have been evaluated by means of the following indexes, both computed over the 1300 pairs of the test set (the average values are reported in Table 1):

- the symmetric difference Δ_s between the area occupied by the ferrofluid as obtained by solving (24)-(34) and the one provided by the neural network;
- the difference n_c between the number of cells occupied by ferrofluid as obtained by solving (24)-(34) and the one provided by the neural network.

From the results reported in Table 1, it turns out that the approximating capabilities of neural networks increase with the number ν of neurons, as expected. In fact, lower values for the average Δ_s can be observed. Fig. 8 shows two examples of the interfaces between water and ferrofluid at certain randomly-extracted time steps as obtained by solving the system equations (24)-(34) and provided by the neural networks with $\nu = 30$ neurons. One can notice that the two interfaces are similar, thus confirming the small value of the symmetric difference Δ_s, i.e., the good approximating capabilities of the trained neural networks.

To devise suitable control actions, we have solved problem (38) by using the mesh adaptive GPS algorithm provided by the Matlab function *patternsearch*. More specifically, we have adopted the neural network providing the best results in terms of accuracy, i.e., the network with $\nu = 30$ neurons. The regime is reached after $t = 7$ (t is the dimensionless time used in the Navier-Stokes equations).

The results obtained by using the black-box model have been compared with those provided by the numerical solution of equations (24)-(34) over a set made up of 12 different reference shapes. The results in terms of the symmetric difference Δ_s and the time required to find a solution to problem (38) are shown in Table 2. Fig. 9 contains the results of the control for the reference shapes 5 and 8. Similar results have been obtained for the other shapes, but they are not reported for the sake of compactness.

It turns out that the black-box model allows one to save a huge amount of time as compared to the application of the full system equations (24)-(34), at the price of only a slight decay of performance on the average. In fact, an average decay of about 16% of the values of Δ_s is experienced by using neural networks, but a saving of the 99% of computing time is achieved.

Table 2: Results of the optimal control approach.

System model	Black-box model with $\nu = 30$											
Reference shape	1	2	3	4	5	6	7	8	9	10	11	12
Δ_s	94	242	236	189	169	121	94	86	111	248	217	158
Time [s]	590	668	800	600	622	648	616	623	640	628	640	631
System model	Full system equations (24)-(34)											
Reference shape	1	2	3	4	5	6	7	8	9	10	11	12
Δ_s	242	170	165	66	202	94	140	71	95	276	67	54
Time [h]	34	31	32	24	21	26	19	36	26	19	22	35

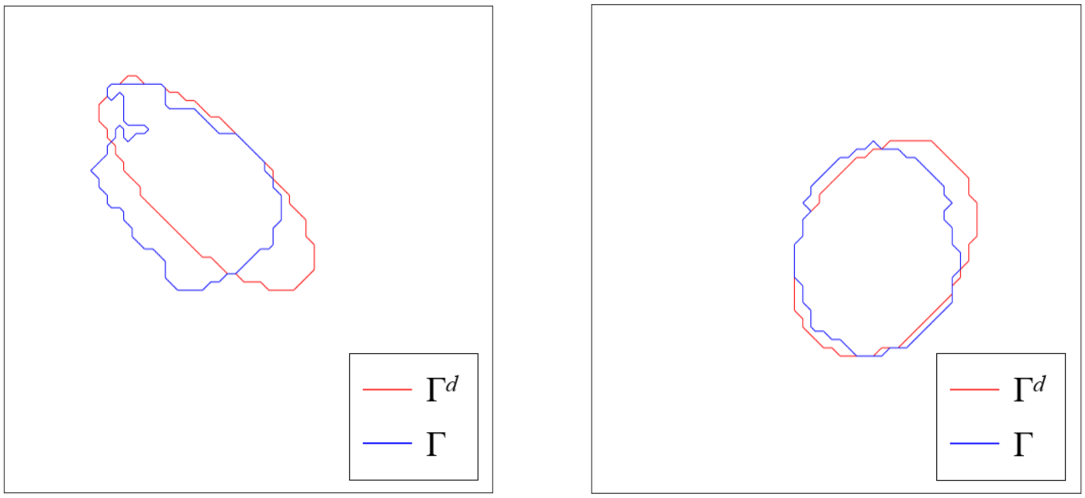

Figure 9: Results of the control actions for the reference shapes 5 (left) and 8 (right).

4 Modeling an Airlift Reactor Process Using Level Sets

Since the seventies, airlift reactors (ALRs) have achieved a lot of success for microorganism growth and cultured plant cells in general. A number of ALRs are employed in bioprocess engineering, depending on the farmed species, from RNA (Ichii, Takehara, Konno, Ishida, Sato, Suzuki, and Yamazumi, 1993) to ethanol production (Vicente, Dluhy, and Teixeira, 2009). Other important fields of application are food industry (Hamada, Ishiyama, and Motai, 1989) and the the production of drugs (Xu, Luo, and Yuan, 2011) and fragrances for cosmetics (Mihaľ, Gavin, and Markoš, 2013). Biodegrading capabilities for wastewater treatment are another field of application (Cozma and Gavrilescu, 2012). The need to process plastic residues has pushed research on bioplastics, where ALRs allows to produce polyesters of hydroxyalkanoates (PHAs) as an alternative to petrochemical polymers (da Silva, Antonio, Rossi, and Pena, 2014). ALRs have been investigated for microalgae nurture to produce biofuels (Chisti, 2007) and/or CO_2 sequestration (Sayre, 2010). In this specific area of application, ALRs can be conveniently incorporated into biorefineries and integrated power generation systems. In this context, we have studied the rising of gas bubbles in a pilot-scale external loop airlift photobioreactor in which microalgae are used to capture CO_2 from flue gas and to treat wastewater. When biological reactants such as microalgae are used, the possibility to manipulate the bubble flow regime during microalgae growth can increase the efficiency of the reactor. In this application, the role that optimal control of level sets (LSs) can play appears to be really appropriate with the goal to impose a process regime in which the average shapes and trajectories of the bubbles are generated in such a way to maximize the efficiency of the reactor (see an example of such reactors in Fig. 10).

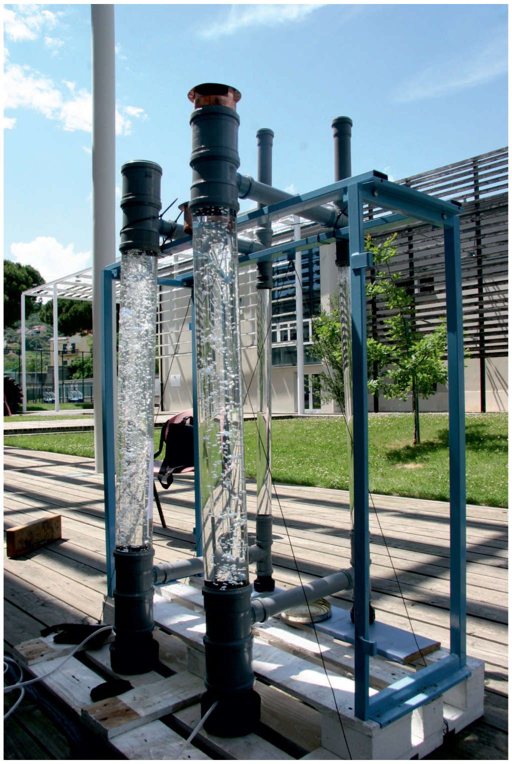

Figure 10: Airlift reactor installed in the Savona Campus of the Genoa University.

4.1 Bubble dynamics in an airlift reactor

The study of the hydrodynamics of the multi-phase flow is crucial for the proper operation of ALRs. The performances of the shape and trajectory of the bubbles in the riser may significantly affect the mass transfer between gas and microalgae and hence their own growth. To construct a model, we have to account for two phenomena as well as their mutual influence: viscous fluid flow and moving interfaces. Fluid flow can be described by Navier-Stokes equations, while interfaces are modelled by using LSs. Toward this end, let us consider the local formulation of the Navier-Stokes equation, obtained from momentum balance by resorting to the divergence theorem in conjunction with the Reynolds transport theorem, (Batchelor, 2000; Chorin and Marsden, 1990):

$$\rho \, \frac{\mathrm{D}u(x,t)}{\mathrm{D}t} = -\nabla p(x,t) + \left(\zeta + \frac{1}{3}\mu\right) \nabla \left(\nabla \cdot u(x,t)\right) + \mu \, \Delta u(x,t) + \xi(x,t) \tag{39}$$

where ρ is the density of the fluid; $u(x,t)$ is the velocity and $\mathrm{D}u(x,t)/\mathrm{D}t$ its material derivative; $p(x,t)$ is the pressure; μ is the dynamic viscosity; ζ is the volume viscosity (second viscosity coefficient); $\xi(x,t)$ accounts for body forces (per unit volume) such as the gravitational force.

The continuity equation holds for Newtonian, incompressible fluids and hence the velocity field is a solenoidal vector field, i.e.,

$$\nabla \cdot u(x,t) = 0\,.$$

Moreover, considering the gravitational force $g(x,t)$ and hence $\xi(x,t) = \rho \, g(x,t)$, we obtain

$$\frac{\mathrm{D}u(x,t)}{\mathrm{D}t} = -\frac{1}{\rho} \, \nabla p(x,t) + g(x,t) + \frac{\mu}{\rho} \, \Delta u(x,t) = -\frac{1}{\rho} \, \nabla p(x,t) + g(x,t) + \nu \, \Delta u(x,t) \tag{40}$$

where $\nu := \mu/\rho$ is the kinematic viscosity. In order to get the nondimensionalized Navier-Stokes equation, let us denote by L, U, and G the characteristic length, characteristic speed, and gravitational acceleration, respectively. Thus, let us define the non-dimensional variables

$$x^* := \frac{x}{L} \quad u^*(x,t) := \frac{u(x,t)}{U} \quad g^*(x,t) := \frac{g(x,t)}{G} \quad p^*(x,t) := \frac{p(x,t)}{\rho \, U^2}$$

and hence it is straightforward to refer to a suitable time scale with

$$T := \frac{L}{U}$$

denoting the convective time. Based on such definitions, (40) can be rewritten as follows

$$\frac{\mathrm{D}u^*(x,t)}{\mathrm{D}t} = -\nabla^* p^*(x,t) + \frac{g\,L}{U^2} \, g^*(x,t) + \frac{\nu}{L\,U} \, \Delta^* u^*(x,t) \tag{41}$$

where $\nabla^* := L \, \nabla$ and $\Delta^* := L \, \Delta$. Two well-known similarity numbers appear in such a model formulation, i.e., the Reynolds and Froude numbers defined as follows:

$$Re := \frac{L\,U}{\nu} \quad Fr := \frac{U}{\sqrt{gL}}\,.$$

More specifically, from (41) we obtain

$$\frac{\mathrm{D}u^*(x,t)}{\mathrm{D}t} = \frac{\partial u^*(x,t)}{\partial t} + u^*(x,t) \cdot \nabla^* u^*(x,t) = -\nabla^* p^*(x,t) + \frac{1}{Fr^2} \, g^*(x,t) + \frac{1}{Re} \, \Delta^* u^*(x,t)\,. \tag{42}$$

Based on the aforesaid, the link between the equations for the velocity field subject to (42) and the interface tracking based on LSs is offered by the formulation of the physical properties of the considered fluids as functions of $\phi(x,t)$. Toward this end, let us consider some smoothed Heaviside function H_ε such as

$$H_\varepsilon(\phi) := \begin{cases} 0 & \phi < -\varepsilon \\ \dfrac{1}{2} + \dfrac{\phi}{2\varepsilon} + \dfrac{1}{2} \sin\left(\dfrac{\pi\phi}{\varepsilon}\right) & -\varepsilon \leq \phi \leq \varepsilon \\ 1 & \phi > \varepsilon \end{cases}$$

where $\varepsilon > 0$ is a parameter affecting the numerical smearing and usually of the same order of magnitude as the interface thickness (Osher and Fedkiw, 2003; Sharma, 2015).

Owing to the interface, we need to redefine the density ρ and kinematic viscosity ν of the two-phase flow system of incompressible fluids as follows:

$$\begin{aligned} \rho &= \rho_g + (\rho_l - \rho_g) H_\varepsilon(\phi) \\ \nu &= \nu_g + (\nu_l - \nu_g) H_\varepsilon(\phi) \end{aligned}$$

where the subscripts g and l tags refer to the gas and liquid phases, respectively. Another ingredient to model multiphase flows is the surface tension force (per unit length), which will be denoted by ξ_{ST}. In the context of front-tracking methods, the most widespread way to take into account such parameter is as a force concentrated on the interface (Brackbill, Kothe, and Zemach, 1992), i.e.

$$\xi_{\mathrm{ST}} = \sigma \kappa n$$

where σ is the coefficient of surface tension,

$$n(x,t) := \frac{\nabla \phi(x,t)}{|\nabla \phi(x,t)|}$$

is the normal to the moving interface, $\kappa(x,t) := \nabla \cdot n(x,t)$ is the curvature, and, for computational purposes, the "mollified" delta function $\delta_\varepsilon(\phi)$ defined as the distribution derivative of $H_\varepsilon(\phi)$ is used instead of the Dirac delta $\delta(\phi)$.

Thus, after adding the term of surface tension to (42) it follows that

$$\frac{\partial u^*(x,t)}{\partial t} + u^*(x,t) \cdot \nabla^* u^*(x,t) = -\nabla^* p^*(x,t) + \frac{1}{Fr^2}\, g^*(x,t) + \frac{\kappa(x,t)\, n(x,t)\, \delta_\varepsilon(x,t)\, L^2}{We} + \frac{1}{Re}\, \Delta^* u^*(x,t) \tag{43}$$

where

$$We := \frac{\rho\, L\, U^2}{\sigma}$$

is the Weber number. The resulting model is illustrated in Fig. 11.

4.2 Numerical schemes for simulation

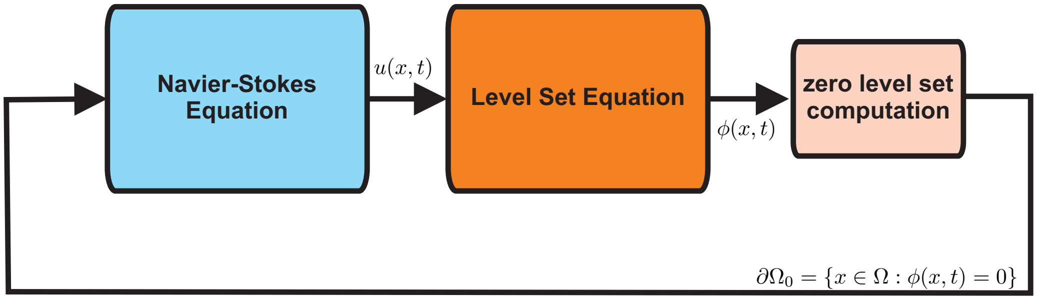
Navier-Stokes Equation
$u(x,t)$
Level Set Equation
$\phi(x,t)$
zero level set computation
$\partial\Omega_0 = \{x \in \Omega : \phi(x,t) = 0\}$

Figure 11: Numerical simulation scheme.

The bidimensional non-dimensional Navier-Stokes equation is solved by employing Chorin's projection on a marker-and-cell (MAC) staggered grid (see Fig. 12). The relative code is based on the Navier-Stokes solver programmed by Seibold (2008a), which relies on a three-steps semi-implicit scheme for time discretization (explicit treatment of the nonlinear convective term, implicit handling of the diffusive term and pressure correction). The portion concerning the resolution of the

Hamilton-Jacobi equation for displaying the zero LS can be instead developed by using Mitchell's toolbox (Mitchell, 2008) as a mold. Specifically, the spatial discretization is performed by the means of an upwind second-order essentially non-oscillatory (ENO) scheme whilst time discretization is carried out with a 3-steps, second-order total variation diminishing (TVD) Runge-Kutta scheme. The choice of upwind approximations is motivated by the hyperbolic nature of the PDE, since in this way numerical instabilities may be avoided. All the PDEs are treated by relying on finite difference approximations.

The equations are numerically solved in the spatial domain $\Omega = [-d_r/2, d_r/2] \times [-L_r/2, L_r/2]$, where L_r is the length of the ALR riser, equal to 0.78 m, and d_r is its diameter, equal to 0.11 m. Ω is discretized to form a mesh grid of rectangular cells, made up of 300 nodes in each dimension. Another possibility consists in focusing on the bottom section of the airlift reactor, with the diffuser diameter (equal to the riser diameter) as the characteristic length. Since the collector diameter d_c is about half the riser diameter ($d_c = 0.05$ m), in this case a square domain could better capture both bubble shape modifications and initial behavior in the inlet section, where the diffuser is highly influenced by the lateral collector.

Concerning the first instance, i.e., Ω as rectangular domain, the boundary conditions employed to evaluate the velocity field are of the Dirichlet type. No-slip conditions are imposed on three sides of the domain. The right-side wall, on the contrary, is modelled as to consider the inflow and outflow determined by the presence of the horizontal collectors. Hence, in correspondence of the openings, a noise-imposed, fairly flat profile velocity distribution typical of the turbulent fluid flow is adopted, while no-slip conditions are used elsewhere.

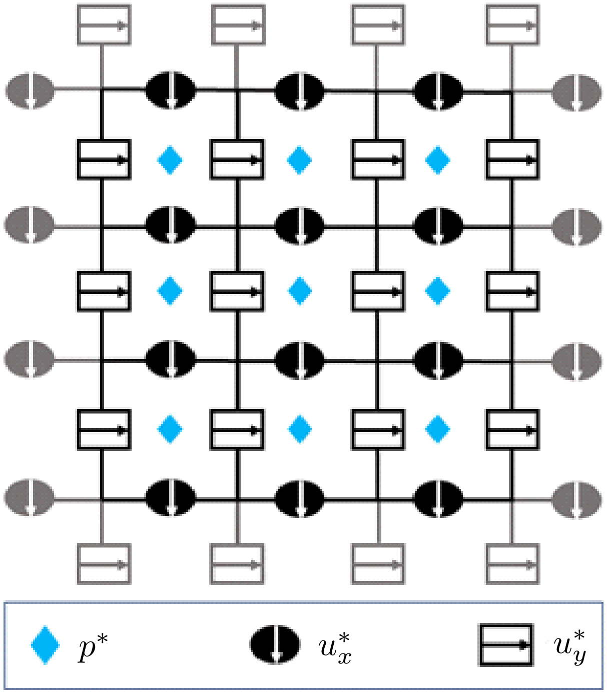

Figure 12: Staggered grid with boundary cells.

4.3 Simulations and validation

The ALR dimensions can be summarized as follows. As previously said, the riser has length and diameter equal to 0.780 m and 0.110 m, respectively. For the same dimensions of downcomer, we have 0.780 m and 0.050 m. The length and diameter of the horizontal collectors are 0.385 m and 0.050 m, respectively. Temperature, pressure, and NaCl pulse injection are taken equal 293 K, 3-4 bar, and 1-10 g L^{-1}, respectively. Other experimental variables are the gas velocity (0.57-0.64 m s^{-1}); liquid velocity in the riser (0.06-0.14 m s^{-1}); liquid velocity in the downcomer (0.17-0.39 m s^{-1}). The minimum, average, and maximum diameters of the bubbles are equal to 0.003 m, 0.007 m, 0.020 m, respectively; the number of bubbles are in the range between 220 and 2800. The remaining physical parameters have been chosen according to standard values: specific CO$_2$ gas constant (188.9 J kg^{-1} K^{-1}), CO$_2$ surface tension coefficient (72.86 $\cdot$ 10^{-3} N m^{-1}), water density (1000 kg m^{-3}), water dynamic viscosity (1.002 $\cdot$ 10^{-3} Pa s), and CO$_2$ dynamic viscosity (1.47 $\cdot$ 10^{-5} Pa s)

In order to maximize mass transfer the maximum superficial area of exchange is required. Also, it is self-evident that the longer the transport phenomenon lasts, the better. Referring this notion to the problem under examination, this means that, in addition to requiring suitable forms of the bubbles, it is also necessary to optimize their residence time in the riser or, in other words, to ensure that appropriate trajectories are achieved. Hydrodynamics of multiphase flow thus holds a controlling influence on mass transport phenomena. As long as the gas inlet velocity is maintained below a threshold value, dependent on the tube geometry, bubbles rise almost individually without significant interactions between them and with narrow bubble size distribution. In this flow condition, known as bubble or homogeneous flow, values of the diameter of the bubble d_b (taken equal to the diameter of a sphere having the same volume as the bubble) generally fall within the range 1-7 mm. The ascent path is mostly rectilinear, with minor transverse and axial oscillations (Abdulmouti, 2014). Whenever the gas phase velocity exceeds the aforementioned threshold, the density of the gaseous fraction in the liquid gradually increases, resulting in a greater interaction between the bubbles, with collisions, clusters formation and the occurrence of coalescence phenomena. The consequential appearance of larger bubbles significantly alters the hydrodynamic scenario, with the concomitant presence of large (more than 20 mm) and small bubbles. These latter bubbles rise rather fast (1-2 m s^{-1}) stirring the liquid. The name of churn flow (also known as heterogeneous flow) is due to the fact that the larger bubbles tend to churn up the liquid (Krishna and Baten, 2001). In this state, as the corresponding Reynolds numbers prove to be higher, spiraling and zigzagging motions can be observed. Moreover, due to this rather turbulent environment, large bubbles often do not count on a clear definition of their form which rather fluctuates quite casually. Nevertheless, some characteristic shapes can be identified, since the morphology of the bubbles is particularly a function of the diameter, speed and properties of the system. The work carried out by Grace (173) produced a well-known generalized graphical correlation (see Fig. 13) that depicts the individual geometry of a single rising bubble in terms of three dimensionless numbers: Reynolds number, Eötvös number (Eo), and Morton number (Mo), i.e.,

$$Eo = \frac{g(\rho_l - \rho_g)d_b^2}{\sigma} \quad Mo = \frac{g\mu_l^4(\rho_l - \rho_g)d_b^2}{\rho_l^2\sigma^3}$$

where μ_l is the viscosity of the liquid and the characteristic length required to compute Reynolds number is d_b.

The diagram of Fig. 13 shows that the preferable shape is that of a spherical cap, given the high ratio between the exchange surface and the occupied volume. Proceeding with the reasoning, a zigzagging trajectory seems to be more suitable, as it would extend the permanence time of the bubbles in the riser. The experimental survey reported bubbles having an average mean diameter of 7 mm with shapes that change from spherical, especially at the bottom of the riser (as it is to be expected since the orifices from which gas is introduced are circular), to spherical caps. In Fig. 14 a snapshot of bubbles generated during one of an experimental test are shown. The computational model has been developed by gradually increasing the complexity of the simulated system: at first we have followed a single bubble, with the final aim of considering multiple puffs of bubbles inserted in the domain at different times.

Owing to the well-known volume loss problem by which LS methods are intrinsically affected, it has been necessary to introduce a correction apt to remedy such an issue. With a mere relocation of

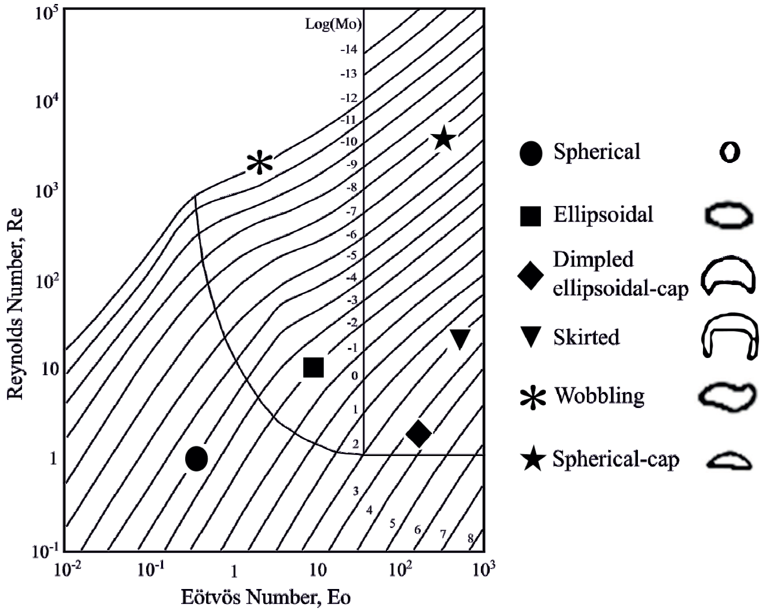

Figure 13: Grace's diagram.

the LS function, lowered or raised according to need, using a simple algorithm based on the bisection method, the simulation resulted more than satisfactory, as it can be noticed in Fig. 15.

Substantially, one can exploit the fact that, provided that the scheme for the convection of ϕ is sufficiently accurate, the error on the volume balance in each time step should be very small. In order to avoid the error accumulation, which instead leads to the significant observed losses, it is therefore advisable to apply the correction at each time step, at the same time paying attention not to alter the shape of the front. Presupposing a suitable rate of reinitialization of ϕ, in the proximity of the front, the LS function is the signed distance and therefore, in the vicinity of its zero iso-contour line, ϕ presents LSs approximately equidistant to each other. Then, by translating ϕ upward or downward by a signed constant K, which represents the distance between the original LS and the one after the translation, the volume (or area in two dimensions) occupied by the gas is conserved and the shapes of the interfaces are essentially unaltered (Fig. 16).

In order for this method to be reliable, K must be small (Smolianski, 2001). The algorithm counts the number of cells inside the interface at two successive times in order to judge if the latter has increased or decreased and therefore if it is necessary to lower or raise the zero LS. The value of K is expected to be between zero and the maximum of the zero LS, so to identify it, one possibility is to use the bisection method within these two extremes.

Although the precaution of inserting a correction on the balance of front volume worked just fine for the case of a single bubble, a further problem of distribution many occur in that of multiple bubbles: the number of grid cells standing within the fronts is preserved but the lost cells are typically added to the larger bubbles, without taking into account their position. The effect that follows is a sort of instantaneous transfer of matter, obviously devoid of physical meaning. It is possible to circumvent this issue as long as one simulates the evolution of multiple bubbles that begin their walk at the same time and the frequency with which the correction is performed is suitably reduced, as shown in Fig. 17. Nevertheless, since this route is not feasible, it is necessary to implement a different approach. As a consequence, the adoption of the so-called conservative LS methods has been adopted (Olsson and Kreiss, 2005; Olsson, Kreiss, and Zahedi, 2007).

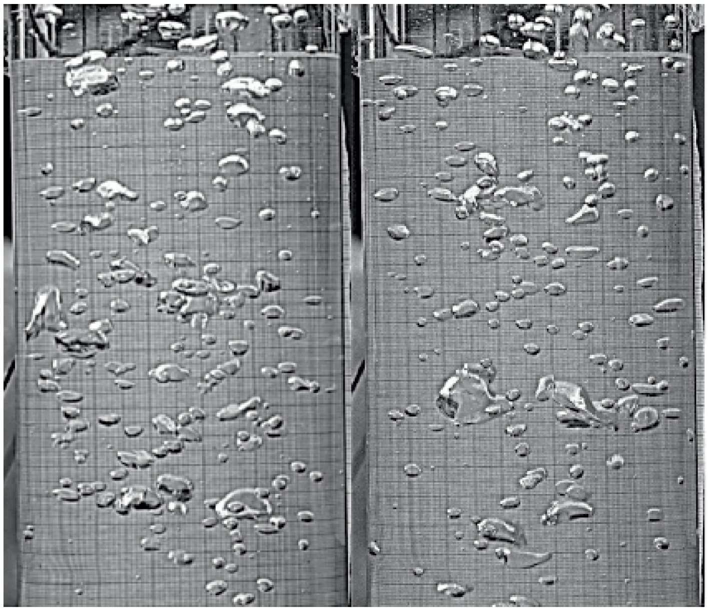

Figure 14: Formation of bubbles in the riser.

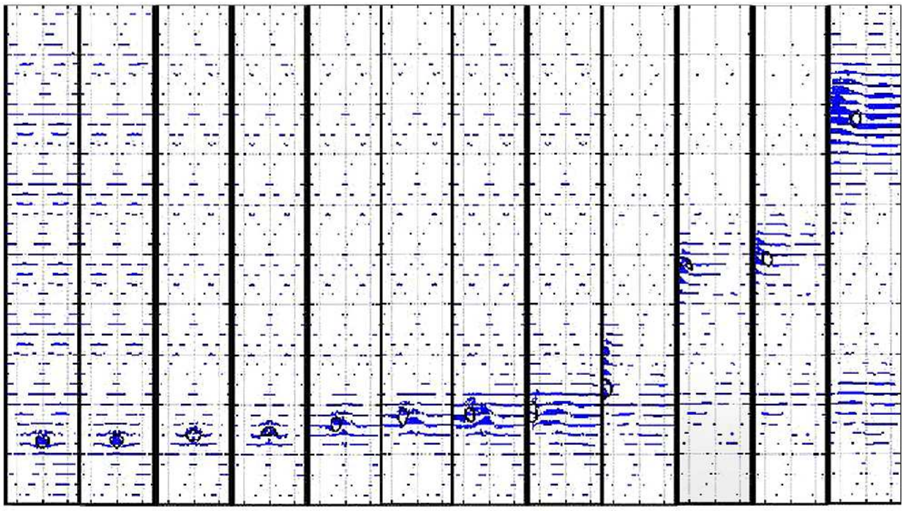

Figure 15: Simulation of a single bubble dynamic in the riser.

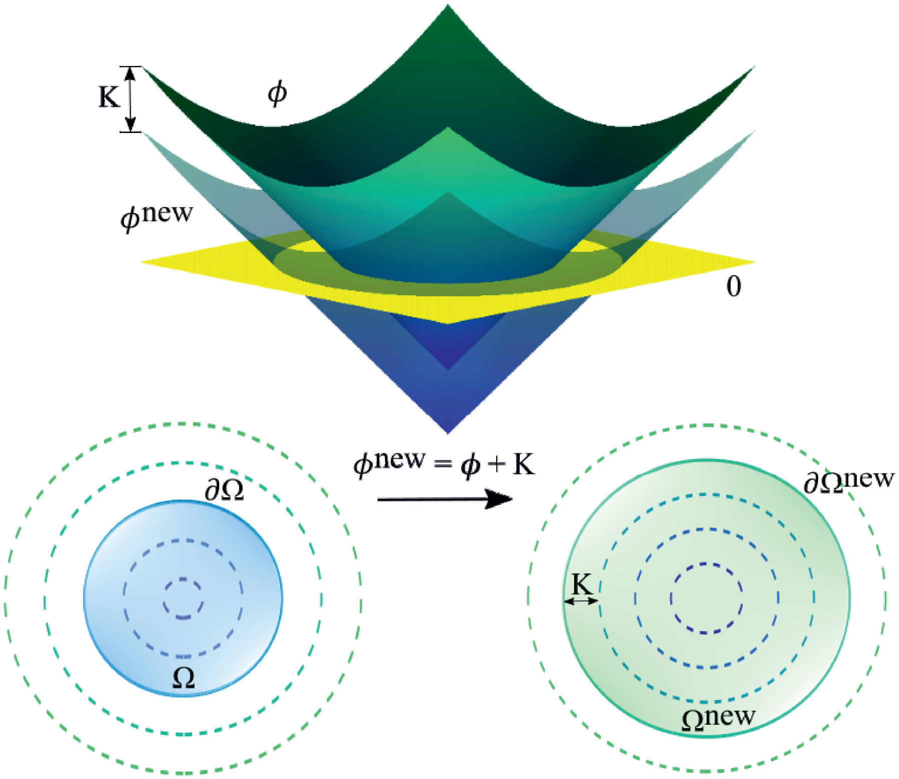

Figure 16: Level set correction.

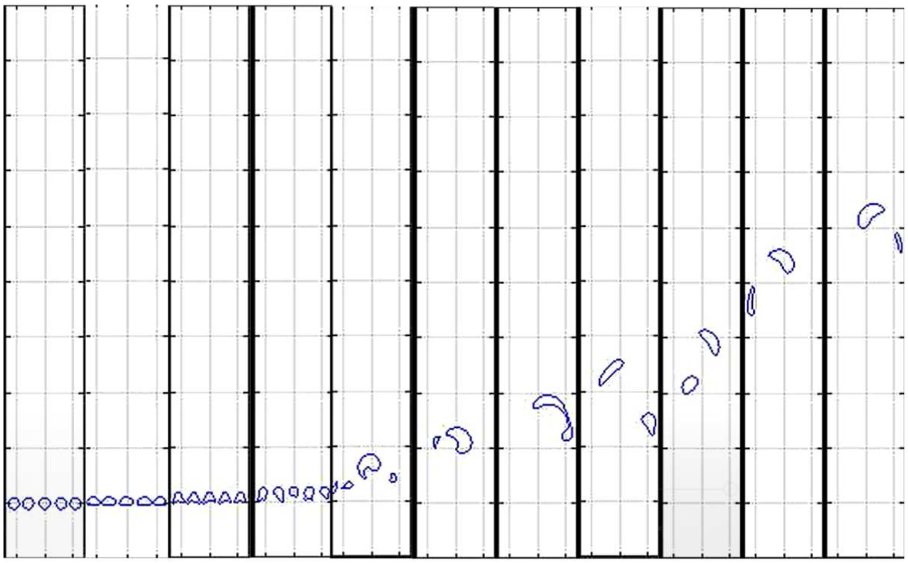

Figure 17: Simulation of an instantaneous puff behavior for a 5 bubbles flow inside the riser with volume correction.

5 Conclusions

Thanks to AFOSR, we have attacked difficult but challenging problems with novel ideas and working together with a large number people. We have mixed theoretical understanding of such problems and experimental activities, as shown with a large of publications. Three PhD students have been involved in the project. Giulia Fabrini has got her PhD in June, 2017. Luca Mantelli is continuing his second year of PhD and he is still working on the first test rig. Matteo Neviani's PhD has concerned the second experimental setup. He will defend his PhD within June 2019.

In the prospect of future investigation, the following goals will be considered to go on with the job we are still completing to disseminate the large extent of experiences gained in these three years of intensive work.

1. We have realized the importance that the control of level sets can play in the research area of fluid control. We plan to investigate more and more the control of multiphase fluid systems modeled by the Navier-Stokes equations, where very little can be found in the literature of the research area of control theory. Indeed, there exists a large literature in the area of computational fluid dynamics (CFD) concerning such topics but with a poor knowledge of pretty well-established tools such as those based on convex optimization.

2. To bridge this gap, we would like to involve leading researchers in CFD. In relation also with the previous observation, it is worth noting that very little can be found in the literature that involve both control and CFD. For example, a simple check concerning the last American Control Conference in Milwaukee shows that only 14 presentations over 1725 includes the keyword CFD. This ratio is even worse in the case of the 57th IEEE Conference on Decision and Control (held in Miami Beach, December 2018), where one can find the acronym CFD in only 5 papers over 2063.

3. The control and estimation of parameter distributed systems is really a challenge, especially when one has to deal with model described by nonlinear partial differential equations. Moreover, such systems are not sufficiently known in many areas of the industrial and mechanical engineering, thus another goal may be that of building a laboratory aimed at bridging this gap by making student interested to all the subjects related to parameter distributed systems and industries fully aware of the competitive advantage achievable by investing in the research on this topic.

6 Acknowledgments

We wish to thank Fariba Fahroo and Shad Reed for their support and suggestions.

References

H. Abdulmouti. Bubbly two-phase flow: Part I- characteristics, structures, behaviors and flow patterns. *American Journal of Fluid Dynamics*, 4(4):194–240, 2014.

A. Alessandri, M. Cuneo, S. Pagnan, and M. Sanguineti. A recursive algorithm for nonlinear least-squares problems. *Computational Optimization and Applications*, 38(2):195–216, 2007.

A. Alessandri, P. Bagnerini, and M. Gaggero. Optimal control of PDE-based systems by using a finite-dimensional approximation scheme. In *Proc. American Control Conf.*, pages 1278–1283, 2013.

A. Alessandri, P. Bagnerini, and M. Gaggero. Optimal control of level sets dynamics. In *Proc. American Control Conference*, pages 2208–2213, 2014.

A. Alessandri, P. Bagnerini, M. Gaggero, and A. Traverso. Further results on the optimal control of fronts generated by level set methods. In *Proc. American Control Conference*, pages 5225–5230, 2016.

A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory*, 39(3):930–945, 1993.

G.K. Batchelor. *An Introduction to Fluid Dynamics.* Cambridge University Press, 2000.

D.P. Bertsekas. Incremental least–squares methods and the extended Kalman filter. *SIAM J. on Optimization*, 6(3): 807–822, 1996.

D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 2nd edition, Belmont, MA, 1999.

J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335 – 354, 1992.

A.K. Caglayan and R.E. Lancraft. A separated bias identification and state estimation algorithm for nonlinear systems. *Automatica*, 19(5):561–570, 1983.

E.E. Callaghan and S.H. Maslen. *The magnetic field of a finite solenoid.* NASA technical note. National Aeronautics and Space Administration, 1960. URL `https://books.google.it/books?id=XIOrBNOeYeoC`.

I. Capuzzo-Dolcetta and P.L. Lions. Hamilton-Jacobi equations with state constraints. *Trans. Amer. Math. Soc.*, 318 (2):643–683, 1990.

Y. Chisti. Biodiesel from microalgae. *Biotechnology Advances*, 25(3):294 – 306, 2007.

A.J. Chorin and J.E. Marsden. *A Mathematical Introduction to Fluid Mechanics.* Springer-Verlag, New York, 1990.

P. Cozma and M. Gavrilescu. Airlift reactors: applications in wastewater treatment. *Environmental Engineering Management Journal*, 11(8):1505–1515, 2012.

D.A. da Silva, R.V. Antonio, J.M. Rossi, and R. da S. Pena. Production of medium-chain-length polyhydroxyalkanoate by Pseudomonas oleovorans grown in sugary cassava extract supplemented with andiroba oil. *Food Science and Technology*, 34:738 – 745, 12 2014.

N. Derby and S. Olbert. Cylindrical magnets and ideal solenoids. *American Journal of Physics*, 78(3):229–235, 2010.

M. Falcone and R. Ferretti. *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations.* SIAM, 2014.

S. Fornaro, S. Maniglia, and G. Metafune. Spazi di funzioni Hölderiane. In S. Fornaro, S. Maniglia, and G. Metafune, editors, *Equazioni ellittiche del secondo ordine*, volume 4, pages 51–68. Edizioni del Grifo, 2004.

K. Fukumizu and S. Amari. Local minima and plateaus in hierarchical structures of multilayer perceptrons. *Neural Networks*, 12:317–327, 2000.

P. Ghosh. *Colloid and interface science.* PHI Learning Pvt. Ltd., 2009.

J.R. Grace. Shapes and velocities of bubbles rising in infinite liquids. *Transactions of the Institution of Chemical Engineers*, 51:116–120, 173.

T. Hamada, T. Ishiyama, and H. Motai. Continuous fermentation of soy sauce by immobilized cells of zygosaccharomyces rouxii in an airlift reactor. *Applied Microbiology and Biotechnology*, 31(4):346–350, 1989.

S. Haykin. *Neural Networks, A Comprehensive Foundation.* Pearson Prentice Hall, Singapore, 2005.

T. Ichii, S. Takehara, H. Konno, T. Ishida, H. Sato, A. Suzuki, and K. Yamazumi. Development of a new commercial-scale airlift fermentor for rapid growth of yeast. *Journal of Fermentation and Bioengineering*, 75(5):375 – 379, 1993.

Y. Iiguni, H. Sakai, and H. Tokumaru. A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter. *IEEE Trans. Signal Proc.*, 40(4):959–966, 1992.

R. Ilin, R. Kozma, and P.J. Werbos. Beyond feedforward models trained by backpropagation: a practical training tool for a more efficient universal approximator. *IEEE Trans. Neural Networks*, 19(6):929–937, 2008.

V. Kůrková and M. Sanguineti. Comparison of worst-case errors in linear and neural-network approximation. *IEEE Trans. Inf. Theory*, 48(1):264–275, 2002.

J.Y. Keller and M. Darouach. Two-stage Kalman estimator with unknown exogenous inputs. *Automatica*, 35(2): 339–342, 1999.

R. Kimmel. *Numerical Geometry of Images.* Springer-Verlag, New York, 2004.

R. Krishna and J.M. Van Baten. Scaling up bubble column reactors with the aid of CFD. *Chemical Engineering Research and Design*, 79(3):283–309, 2001.

A.J. Kurdila and M. Zabarankin. *Convex Functional Analysis.* Birkhäuser Verlag, Basel, Switzerland, 2005.

R.M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

M.G. Crandall and P.L. Lions. Hamilton-Jacobi equations in infinite dimensions I. Uniqueness of viscosity solutions. *J. Funct. An.*, 62(3):379–396, 1985.

M.G. Crandall and P.L. Lions. Hamilton-Jacobi equations in infinite dimensions II. Existence of viscosity solutions. *J. Funct. An.*, 65(3):368–405, 1986.

M. Mihaľ, S.P. Gavin, and J. Markoš. Airlift reactor — membrane extraction hybrid system for aroma production. *Chemical Papers*, 67(12):1485–1494, 2013.

I.M. Mitchell. A toolbox of level set methods (version 1.1) ubc cs tr-2007-11. Technical report, University of British Columbia, Canada, 2007.

I.M. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *J. Scientific Comput.*, 35(2-3):

300–329, 2008.

H.Y. Mussa and R.C. Glen. Memory-efficient fully coupled filtering approach for observational model building. *IEEE Trans. Neural Networks*, 21(4):680–686, 2010.

K. Nishiyama and K. Suzuki. H_∞–learning of layereed neural networks. *IEEE Trans. Neural Networks*, 12(6):1265–1277, 2001.

E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *J. of Computational Physics*, 210(1): 225–246, 2005.

E. Olsson, G. Kreiss, and S. Zahedi. A conservative level set method for two phase flow II. *J. of Computational Physics*, 2225(1):785–807, 2007.

S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003. ISBN 0-387-95482-1.

M. Petit, A. Kedous-Lebouc, Y. Avenas, M. Tawk, and E. Artega. Calculation and analysis of local magnetic forces in ferrofluids. *Przeglad Elektrotechniczny*, 9B, 2011.

K. Reif, S. Günter, E. Yaz, and R. Unbehauen. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Trans. Automatic Control*, 44(4):714–728, 1999.

R. Sayre. Microalgae: The potential for carbon capture. *BioScience*, 60(9):722–727, 2010.

C. Scherer and A.M. Figueiredo Neto. Ferrofluids: properties and applications. *Brazilian Journal of Physics*, 35(3A): 718–727, 2005.

B. Schottky and D. Saad. Statistical mechanics of EKF learning in neural networks. *Journal of Physics A: Mathematical and General*, 32(9):1605–1621, 1999.

B. Seibold. A compact and fast matlab code solving the incompressible Navier-Stokes equations on rectangular domains. Technical report, MIT, Cambridge, 2008a.

B. Seibold. A compact and fast Matlab code solving the incompressible Navier-Stokes equations on rectangular domains mit18086 navierstokes.m. Technical report, 2008b.

J.A. Sethian. *Level Set Methods and Fast Marching Methods*, volume 3. Cambridge University Press, Cambridge, second edition, 1999.

A. Sharma. Level set method for computational multi-fluid dynamics: A review on developments, applications and analysis. *Sadhana*, 40(3):627–652, May 2015.

A. Smolianski. *Numerical modeling of two-fluid interfacial flows.* PhD thesis, Ph.D. Dissertation, University of Jyväskylä, Finland, 2001.

H. J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input–output map. *Neural Networks*, 5:589–593, 1992.

A.A. Vicente, M. Dluhy, and J.A. Teixeira. Increase of ethanol productivity in an airlift reactor with a modified draught tube. *The Canadian Journal of Chemical Engineering*, 77(3):497–502, 2009.

Y. Xu, L. Luo, and J. Yuan. CFD simulations to portray the bubble distribution and the hydrodynamics in an annulus sparged air-lift bioreactor. *The Canadian Journal of Chemical Engineering*, 89(2):360–368, 2011.

I. Yang and C.J. Tomlin. Identification of surface tension in mean curvature flow. In *Proc. American Control Conference*, pages 3290–3295, 2013.

R. Zoppoli, M. Sanguineti, and T. Parisini. Approximating networks and extended Ritz method for the solution of functional optimization problems. *J. Optim. Theory Appl.*, 112(2):403–440, 2002.