UNIVERSITY OF TRENTO

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE
Corso di Laurea Magistrale in Ingegneria Meccatronica

Modelling and optimal control of laser-driven thermocapillary motion of a droplet

Candidato:
Alberto De Marchi

Relatori:
Prof. Francesco Biral
Prof. Enrico Bertolazzi

Correlatori:
Prof. Paolo Bosetti
Prof. Martin M. Hanczyc

Anno Accademico 2015–2016

Alberto De Marchi: *Modelling and optimal control of laser-driven thermocapillary motion of a droplet*, Tesi di Laurea Magistrale, October 2016.

If I have seen further
it is by standing on the sholders of Giants.

— Isaac Newton, 1676

Abstract

The aim of the present thesis is to analyse the migration of a droplet suspended in another liquid in a flat open container and to explore and exploit different strategies to control the motion of the droplet. In particular, the actuation mechanism is the thermocapillary effect, induced by temperature gradients generated by a focused laser beam.

A mathematical model representing the system dynamics is presented, which has been employed with different purposes, from the dynamic simulation up to the synthesis of a controller. Two models are proposed to take into account the thermal dynamics of the suspending liquid, based on a semi-analytical solution and finite elements method, respectively. The model presents characteristic parameters that should be evaluated, therefore the identification problem is discussed and accompanied by experimental results. Also the state estimation is a key step when the controller has to be implemented in practice. About this, estimation algorithms are suggested and analysed. The optimal control theory is utilised to address the considered control problems, which are the minimum time and the reference tracking problems. Both trajectory optimization and model-predictive control are investigated and promising results have been highlighted.

This work aims then at marking the beginning of optimal control in droplet manipulation, as well as at representing a potential guideline for coping with similar projects.

Sommario

Lo scopo di questa tesi è di analizzare il moto di una goccia sospesa in un altro liquido in un contenitore aperto e piano e di esplorare e applicare diverse strategie di controllo del moto della goccia. In particolare, il meccanismo di attuazione è basato sull'effetto termocapillare, indotto dai gradienti termici generati da un fascio laser focalizzato.

Viene presentato un modello matematico che rappresenta la dinamica del sistema; questo è stato adottato per svariate applicazioni, dalla simulazione alla sintesi del controllore. Due modelli sono stati proposti per considerare il comportamento termico del liquido, basati su una soluzione semi-analitica o sul metodo degli elementi finiti. I parametri caratteristici del modello devono essere valutati, pertanto il problema della loro identificazione è discusso e accompagnato dai risultati sperimentali. Anche la stima dello stato del sistema è un passaggio cruciale per poter implementare realmente il controllore. Per questo alcuni algoritmi di stima vengono suggeriti e analizzati. Infine la teoria del controllo ottimo viene adottata per affrontare i problemi di controllo in esame, che sono i problemi di tempo minimo e inseguimento di percorso. Vengono investigati sia l'ottimizzazione di traiettoria che il controllo ad anello basato sul modello, ottenendo risultati incoraggianti.

Questo lavoro ambisce quindi a segnare l'introduzione del controllo ottimo nella manipolazione di gocce, nonché a rappresentare una possibile linea guida per affrontare progetti simili.

Contents

1 Introduction **1**
1.1 Context . . . 2

2 Modelling **5**
2.1 Robot dynamics . . . 6
2.2 Laser-liquid interaction . . . 8
2.3 Temperature field of liquid domain . . . 10
2.3.1 Semi-analytical solution . . . 11
2.3.2 Finite Elements . . . 18
2.4 Droplet and thermocapillary force modelling . . . 21

3 State and parameters estimation **27**
3.1 Experimental setup . . . 28
3.2 Offline parameters identification . . . 30
3.2.1 Data and results . . . 33
3.2.2 Model and parameters validation . . . 35
3.2.3 Steady state identification . . . 36
3.3 Robot dual estimation using Kalman Filter . . . 39
3.4 Droplet dual estimation using Unknown Input Kalman Filter . . . 43
3.5 Liquid dual estimation using Kalman Filter . . . 48

4 Optimal control **53**
4.1 Problem formulation . . . 53
4.2 Methods . . . 57
4.3 OCP with analytical strategy . . . 59
4.3.1 Numerical solution . . . 63
4.3.2 Simulation results . . . 64
4.4 MPC with analytical strategy . . . 68
4.4.1 Numerical solution . . . 70
4.4.2 Simulation results . . . 71
4.5 MPC with FE and direct strategy . . . 74
4.5.1 Numerical solution . . . 78
4.5.2 Simulation results . . . 78

A Analytical and Finite Elements Solution **85**

B Derivation of constraints Jacobian matrix **91**

Chapter 1

Introduction

Optimization techniques have a fundamental impact on current industrial practice. Dynamic processes can be driven by optimization-based feedback controllers, that iteratively solve an optimal control problem (OCP) in order to account for inaccuracies in the mathematical model of the plant and disturbances. Nowadays a variety of approaches and algorithms is available to attack this task [1–7].

A challenging application of optimal control theory is thermocapillary driven droplet manipulation. Droplet-based systems are of growing interest because they can handle multiple discrete samples compared to continuous flow systems [8]. Microfluidic devices that manipulate droplets enable fast large-scale complex biological and chemical assays [9, 10]. Moreover, it has been noticed that droplets can mimic living cells and perhaps possess the fundamentals of intelligent systems [11].

This thesis deals with the design of a model-based optimal controller for droplet manipulation. Droplet-based devices promise a broad range of applications, especially in those fields where cross-contamination and dispersion are not allowed. These devices are tools or platforms that manipulate droplets and hence need various functionalities, e. g. droplet generation, transport, dispensing, merging, mixing, splitting and trapping, and thus different objects like pumps and valves have to be designed [9, 10, 12–16]. The main focus of this work is on, but not restricted to, droplet transport in open field, i. e. without walls restraining the degrees of freedom of droplets. Instead, in microfluidic devices, droplets are typically trapped in microchannels and move when the surrounding continuous phase flows. Many actuation mechanisms have been investigated and cover a broad range of physical principles, including hydrodynamic effects [15], temperature gradients [12, 17], electric and magnetic fields [14], optical and acoustic forces [9, 13] and chemical gradients [18–21]. Droplet transport driven by Marangoni flow—thermocapillary effect— is considered within this work because it is more prominent when the scale becomes smaller as compared to a pressure-driven flow [17]. Furthermore, optical heating is a promising techniques to generate significant localized temperature gradients with limited changes in temperature.

Droplet manipulation through laser-induced thermocapillary forces is a complex task. An optimization-based controller requires a mathematical model of the system. The present work explores a variety of approaches to handle this control problem, and develops a novel control-oriented model that has been experimentally validated.

Finally, based on this dynamic model and different optimality criteria, some control strategies are proposed and tested *in silico*.

Outline In Chapter 2 the elements of the system are defined and a mathematical model suitable for control purposes in developed, based on previous knowledge, literature and experimental evidence. Then Chapter 3 describes the experimental work that has been carried out during the project. Model parameters are identified and validation of the proposed model is also discussed. Moreover, a procedure for online state estimation and simultaneous parameters identification is analysed. In Chapter 4 two control problems are formulated following the approach of optimal control theory. Different strategies to solve these OCPs are suggested and tested by simulations.

1.1 Context

Optimal and model-predictive control Optimal control theory is a highly-developed branch of mathematics. Temperature field are in general governed by the heat equation, that is a PDE. Optimal control of this type of systems or governed by coupled ODEs and PDEs has been investigated [6, 22–25], and the same for systems constrained by PDEs [26]. Various approaches have been proposed to manage PDEs in OCPs, such as full or reduced direct discretization using finite difference schemes [6, 27, 28] or methods based on finite element (FE) discretization [29–34]. Typically from these OCPs large-scale non-linear problems arise, and so techniques like reduced-order modelling (ROM) have been applied to reduce the computational effort [35]. Simplified models are often adopted to deal with linear systems and/or analytical solutions [36]. Optimal control theory is applied in model-predictive control (MPC) of dynamical processes. Many different numerical techniques have been developed to solve OCPs involving large-scale non-linear systems with real-time capabilities [3, 5, 37–40].

Droplet-based systems Droplet-based (digital) microfluidics has drawn much attention as an enabling technology to handle multiple discrete samples with small volume requirement, high throughput and sensitivity, no cross-contamination nor dispersion [9]. The European Commission started an initiative on biochemistry-based information technology (CHEM-IT) [41], that funded projects aimed to integrating chemistry, robotics, artificial life and microbiology. In particular, MATCHIT and EVOBLISS deal with droplets as the link between electronic and mechanical systems and biochemistry to design and exploit chemorobotic systems [42–45]. The National Science Foundation (NSF) recently awarded two projects among others on droplet manipulation, based on thermotaxis and ratchets respectively. The latter developed low-cost devices for parallel microfluidics, where droplet are actuated by the anisotropy of the underlying solid surface [46]. Instead, the first designed a new platform technology on the science of thermotaxis actuation [8, 47, 48]. This demonstrates the interest in lab-on-chip applications and droplet-based systems.

Droplet manipulation based on thermocapillary effect has also been investigated and different technologies have been implemented. For instance, a silicon-based droplet transportation platform with embedded metal micro-heaters has been developed [8], or a planar channel with four micro-heaters embedded in the boundaries [49]. Also the motion along fibers subject to temperature gradients has been experimentally and theoretically proved [50].

Light-induced Marangoni flows can be generated to handle multiple heavy and large objects and fulfill different functionalities [9, 12, 13, 51–53], also with transparent objects [54]. Mode bifurcation and spontaneous symmetry-breaking due to temperature-induced instabilities have been successfully modelled with a set of simple coupled differential equations [55]. An up-to-date review on thermocapillarity in microfluidic applications can be found in [17].

Chapter 2

Modelling

The aim of this Chapter is to define a mathematical model of the system composed by a robot, a laser, a liquid layer and a droplet, Fig. 2.1. These elements are linked and connected through mass, energy or information transfer. The same is valid also in relation to the environment. Moreover, the user may act and have an effect on the system, typically with the purpose of achieving a desired behaviour. We try to build a control-oriented model, in the sense that it should catch the essence of the phenomenon but simple enough to find a control policy. A good model may have few states with a strongly coupled non-linear dynamics or, on the other hand, it may be a large-scale system with linear dynamics.

The contributions of this Chapter are mathematical models that may represent the system under analysis and hopefully describe its behaviour. In particular, they are the following:

- linear dynamical model of a robotic platform;
- thermal model of the laser-liquid interaction;
- semi-analytical solution of the temperature field induced by laser heating on a thin liquid layer;
- finite element (FE) model of a thin liquid layer and application of laser heating;
- point-mass model of a droplet subject to drag and linear thermocapillary force.

These objects compose a high-level model of the overall system, that is experimentally validated in Chapter 3 and adopted for control purposes in Chapter 4. Notice that the semi-analytical solution and the FE thermal model are not employed simultaneously to describe the temperature field.

Robot dynamical model presented in §2.1 is quite general yet simple. It does not refer to the specific robotic platform developed by the EVOBLISS project [43], called EvoBot, but it is a mathematical object that may represent a wider class of mechanical systems, in which EvoBot falls when in normal operative conditions.

Afterwards, a model of laser-liquid interaction is defined in §2.2. It is worth noticing that this is the link between the robotic platform (where the emitter device is) and the liquid layer (where heating is applied). Both assumptions on robot and liquid apply and are taken into account in modelling this link between subsystems. Effect of laser

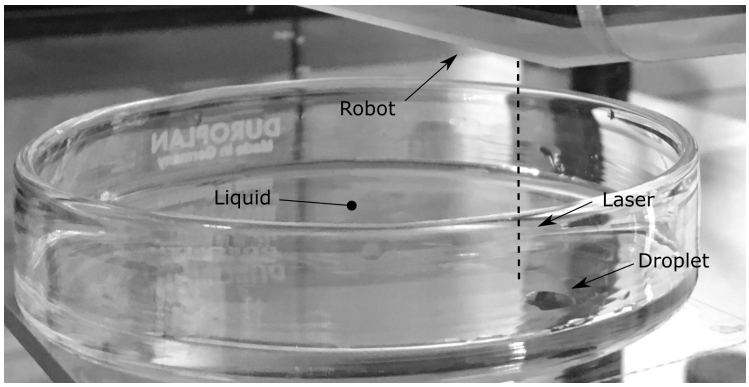


Figure 2.1: Elements of the system: robot head, laser, liquid and droplet.

on temperature field is a key step because it is a filter between control inputs and droplet motion.

Temperature field evolution is a central part in modelling the system because thermal gradient induced by laser heating is responsible for the thermocapillary forces that act on the droplet. Thus, a correct estimation of thermal gradients is important to accurately predict the droplet behaviour (assuming determinism). At this step, we derive and implement two different approaches: the first model is based on linearity of the heat equation and yield to a semi-analytical solution (few states, coupled, nonlinear), the second one is based on Galerkin finite element (FE) approximation and leads to a numerical solution (large-scale, sparse, quasi-linear). These two are presented in §2.3.1 and §2.3.2 respectively. An estimate of thermal gradient in droplet position is derived, based on analytical or numerical methods.

Finally, we formulate a model for thermocapillary motion of droplets in §2.4. This model should represent the physical phenomenon more accurately as droplets become smaller (point mass for thermal gradient and less convective flows, but probably more sensitive to noise). The model is based on classical works on thermocapillary migration of droplets and bubbles and more recent experimental investigations.

It is worth to discuss about model order reduction (MOR). This technique aims at lowering the computational complexity and dimensionality of mathematical models of processes, for example, in simulations of large scale dynamical systems. An approximation to the original model, i. e. a reduced order model, is computed and its evaluation is less costly w.r.t. the original full model [56]. Common approaches are proper orthogonal decomposition (POD), balanced truncation, reduced basis method, matrix interpolation and transfer function interpolation. Recently, MOR has been intensively developed and applications have been found also in optimization and control. In this work MOR is not adopted, but it might be a great step forward when combined with the FE model proposed for modelling the liquid medium.

2.1 Robot dynamics

Robotic platform developed within the EVOBLISS project [43] is based on a modified 3D printer developed, tested and described in [44, 57]; see Fig. 2.2. Essentially it is

composed by two couples of linear guide rods, actuated by stepper motors and timing belts, forming a gantry layout (Cartesian kinematics).

The device emitting laser is attached onto the moving head of EvoBot and it is considered to be rigidly connected to the mechanical system. Within this work position of the laser spot (referred to as laser position) and actual configuration of the robotic platform (referred to as robot position) have the same practical meaning.

The EvoBot is a mechatronic product, where electronic devices sense and control the mechanical structure. Modelling the robotic platform requires to take into account the kinematic relationship between bodies that constitute the system and all the forces applied onto those bodies.

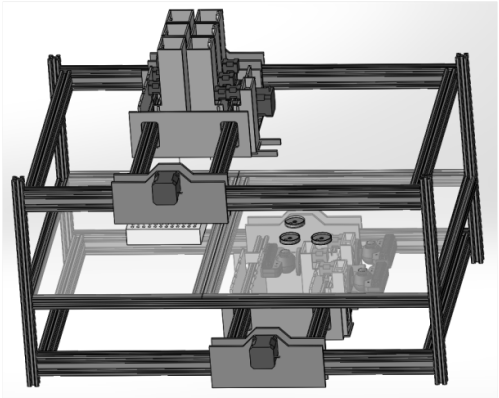


Figure 2.2: Robotic platform used within the EVOBLISS project [43].

Let us define vector $\mathbf{x}$ that identifies the configuration of the system, and its first time-derivative $\mathbf{v} = \dot{\mathbf{x}}$. Entries of $\mathbf{x}$ are the Cartesian coordinate that describe the actual system configuration (thanks to the Cartesian architecture); these are two uncoupled variables, because the laser motion is planar and axes are independent.

The forces that play a role in robot dynamics are due to control inputs and friction. In EvoBot, stepper motors drive the moving head, with a target position and reference velocity profile. Notice that stepper motors involve an inner control loop because they are position-controlled. In this work we do not deal with this specific feature but we consider actions linearly dependent on control inputs. This may be the case with DC motors or stepper motors in non-slipping conditions. Mechanisms of friction may be various, e. g. Coulomb, viscous, Karnopp, Lugre and Dahl models, and may explain different effects, like stick-slip and Stribeck effect. From the second Newton's law we have that for every time t the model reads

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t) \tag{2.1}$$

$$\boldsymbol{M}_L(\mathbf{x}(t))\dot{\mathbf{v}}(t) = \boldsymbol{B}_L(\mathbf{x}(t))\mathbf{u}(t) - \boldsymbol{C}_L(\mathbf{x}(t), \mathbf{v}(t))\mathbf{v}(t) \tag{2.2}$$

where $\boldsymbol{M}_L(\mathbf{x})$ is the configuration-dependent mass distribution, $\boldsymbol{B}_L(\mathbf{x})$ is the input matrix, $\mathbf{u}$ is the control input (e. g. electric signals) and $\boldsymbol{C}_L(\mathbf{x}(t), \mathbf{v}(t))$ accounts for the frictional effects.

Let us make some assumptions for the EvoBot platform. We have that inertia properties do not depend on the configuration, thus $\boldsymbol{M}_L(\mathbf{x}) = \boldsymbol{M}_L$. Moreover, neglect-

ing dynamics of motor drivers we have $\boldsymbol{B}_L(\mathbf{x}) = \boldsymbol{B}_L$. Then, we recall that this work does not focus on control techniques for friction compensation but on thermocapillary motion induced by laser heating. The available mechanical system is only a mean to move the laser beam, and thus we do not deal with friction-related problems in this work; a linear friction model is considered for simplicity and tractability. Notice that the same holds for backlashes and geometrical errors present in the physical system. For the sake of clarity, we repeat that none of these issues is of interest to the aim of this thesis. With these assumptions the model is simplified and reduces to

$$\boldsymbol{M}_L\dot{\mathbf{v}}(t) = \boldsymbol{B}_L\mathbf{u}(t) - \boldsymbol{C}_L\mathbf{v}(t) \tag{2.3}$$

An additional assumption is to consider axes with the same properties, making them indistinguishable. This may be quite accurate for control and friction actions but inertia properties are not equal, because one of the two axes moves also the other. However, we can assume independent equal axes, namely ($\boldsymbol{I}$ is the identity matrix)

$$\boldsymbol{M}_L = m_L\boldsymbol{I}\ , \quad \boldsymbol{B}_L = b_L\boldsymbol{I}\ , \quad \boldsymbol{C}_L = c_L\boldsymbol{I} \tag{2.4}$$

Parameters m_L, b_L and c_L represent mass, input sensitivity and friction coefficient, respectively, of the two axes. Their values identify behaviour and properties of the robotic platform as expected from the mathematical model proposed here.

Summary Let us express the linear dynamical model as a first-order system of ODEs in matrix form. Collecting the robot state in vector $\mathbf{y} = [\mathbf{x}, \mathbf{v}]^T$, the time-invariant model can be written as

$$\boxed{\boldsymbol{M}_y\dot{\mathbf{y}}(t) = \boldsymbol{A}_y\,\mathbf{y}(t) + \boldsymbol{B}_y\,\mathbf{u}(t)} \tag{2.5}$$

where matrices $\boldsymbol{M}_y$, $\boldsymbol{A}_y$ and $\boldsymbol{B}_y$ depend on properties of the robot and are given by:

$$\boldsymbol{M}_y = \begin{bmatrix}\boldsymbol{I} & 0\\ 0 & m_L\boldsymbol{I}\end{bmatrix}\ , \qquad \boldsymbol{A}_y = \begin{bmatrix}0 & \boldsymbol{I}\\ 0 & -c_L\boldsymbol{I}\end{bmatrix}\ , \qquad \boldsymbol{B}_y = \begin{bmatrix}0\\ b_L\boldsymbol{I}\end{bmatrix} \tag{2.6}$$

2.2 Laser-liquid interaction

Let us analyse how to model liquid heating induced by laser absorption. In the absence of any mass transport phenomena, one can find the temperature field $T(t, \mathbf{x})$, $\mathbf{x} \in \Omega \subset \mathbb{R}^2$, by solving the heat equation (energy balance in differential form):

$$\varrho c_p \frac{\partial T}{\partial t}(t, \mathbf{x}) = \nabla_{\mathbf{x}} \cdot (k\nabla_{\mathbf{x}} T(t, \mathbf{x})) + \dot{q}(t, \mathbf{x}) \tag{2.7}$$

where ϱ is the sample specific mass, c_p is the heat capacity of the medium at constant pressure, $\dot{q}$ is the thermal power deposited in the medium per unit volume. Thermal field is supposed to be 2-dimensional (a thin liquid layer with uniform thickness is considered). Assuming that absorbed light is completely and instantaneously converted into heat, the intensity reduction per unit thickness is exactly the heating power per unit volume, namely:

$$\dot{q}(t, \mathbf{x}) = \frac{\Delta I(t, \mathbf{x})}{\Delta z} \tag{2.8}$$

where Δz is the layer thickness; ΔI is the change of light intensity due to the liquid layer, i. e. $\Delta I = I_0 - I$ with incident intensity I_0 and transmitted intensity I.

Let us focus on the absorption of electromagnetic waves, that typically generates localized heating. A beam of light on a material may be scattered, absorbed or transmitted; often these three phenomena occur simultaneously. Let us recall the Beer-Lambert law: the transmitted intensity I at depth z depends on the incident intensity I_0 at $z = 0$ and on the linear attenuation coefficient β_a of the material[1]. If light scattering is negligible, attenuation coefficient matches the absorption coefficient. Notice that these coefficients depend on the light wavelength in general, but here we consider only a monochromatic laser with wavelength λ^* and denote $\beta_a = \beta_a(\lambda^*)$. Thus, for an absorbing layer of thickness Δz along the optical axis, from the Beer-Lambert law, the intensity reduction is given by:

$$I(t,\mathbf{x}) = I_0(t,\mathbf{x})e^{-\beta_a \Delta z} \qquad \Rightarrow \qquad \Delta I(t,\mathbf{x}) = I_0(t,\mathbf{x})\left(1 - e^{-\beta_a \Delta z}\right) \tag{2.9}$$

For a weakly absorbing medium, i. e. $\beta_a \Delta z \ll 1$, the transmitted intensity may be approximated to a first-order truncation of the Taylor expansion, namely:

$$I(t,\mathbf{x}) \approx I_0(t,\mathbf{x})\left(1 - \beta_a \Delta z\right) \qquad \Rightarrow \qquad \Delta I(t,\mathbf{x}) \approx I_0(t,\mathbf{x})\beta_a \Delta z \tag{2.10}$$

Inspired by the result in (2.10), let us introduce parameter β to have the same notation in (2.9) when the assumption on the weakly absorbing medium is not valid. Notice that in the limit $\beta_a \Delta z \to 0$ the definition in (2.11) leads to $\beta = \beta_a$.

$$\Delta I(t,\mathbf{x}) = I_0(t,\mathbf{x})\beta \Delta z \qquad \Rightarrow \qquad \beta = \frac{1 - e^{-\beta_a \Delta z}}{\Delta z} \tag{2.11}$$

So, from the Beer-Lambert law and the definition of β, the heating term in (2.7) is given in general by $\dot{q}(t,\mathbf{x}) = I_0(t,\mathbf{x})\beta$.

Let us discuss now about the light intensity generated by a laser emitter. Laser beam may be modelled as pulsed or continuous in time depending on how radiation is generated. The consequent heating power is influenced also by the photo-thermal conversion, and in particular on its time scale. Thermal diffusion is characterized by the time constant $\tau_{\text{dif}} \sim l^2/\alpha$, having $\alpha = k/(\varrho c_p)$ the thermal diffusivity and l the characteristic length scale of the heat source[2]. Considering a focused probe beam and water as liquid, we may have $l \approx 1 \times 10^{-4}\,\text{m}$ and $\alpha \approx 1 \times 10^{-7}\,\text{m}^2/\text{s}$ and then time scale is about $\tau_{\text{dif}} \approx 0.1\,\text{s}$. Photo-thermal conversion is on a time scale much smaller than τ_{dif}. Thus, in practice, heat release is not delayed w.r.t. incident radiation, and so $\dot{q}(t,\mathbf{x}) = \beta I_0(t,\mathbf{x})$ holds. For example, the heat release due to a laser pulse may be considered instantaneous and modelled with a Dirac delta distribution in time with finite energy density $\varepsilon_0(\mathbf{x})$, namely

$$\dot{q}(t,\mathbf{x}) = \beta \varepsilon_0(\mathbf{x})\delta_D(t) \tag{2.12}$$

[1] The probability of light being absorbed or scattered out of the beam increases with incident intensity I, linear attenuation coefficient $\beta_a(\lambda)$ and thickness z. Integrating and imposing the initial condition $I(x)\Big|_{z=0} = I_0$, the Beer-Lambert law is found.

$$\frac{\mathrm{d}I}{\mathrm{d}z} = -\beta_a(\lambda)I \qquad \Rightarrow \qquad I = I_0 e^{-\beta_a(\lambda)z}$$

[2] Let us introduce dimensionless time $\tau = t/\tau_{\text{dif}}$ and length $\xi = x/l$ into the heat equation with constant homogeneous physical properties. In this way the heat equation gets normalized coefficients, then τ_{dif} and l are characteristic quantities of the system.

$$\frac{\partial T}{\partial t} = \alpha \nabla^2_{\mathbf{x}} T + \frac{\dot{q}}{\varrho c_p} \qquad \Rightarrow \qquad \frac{\partial T}{\partial \tau} = \nabla^2_{\boldsymbol{\xi}} T + \frac{l^2 \dot{q}}{k}$$

Spatial distribution of incident light intensity depends on the transverse profile of the laser beam. This often occurs in the form of Gaussian beam, that is the lowest-degree mode and it remains Gaussian also after passing simple optical elements, therefore it is the most commonly obtained from all stable optical systems [58]. The intensity profile of a Gaussian laser beam centred in $\mathbf{x}_L$ with characteristic radius r_L and overall laser power $p_L(t)$ is described as:

$$I_0(t, \mathbf{x}) = \frac{2p_L(t)}{\pi r_L^2} e^{-2(r/r_L)^2} \qquad r = \|\mathbf{x} - \mathbf{x}_L\| \tag{2.13}$$

As the characteristic radius reduces, the Gaussian distribution approaches the Dirac delta distribution and eventually the point-wise laser beam is obtained for $r_L \to 0^+$, namely

$$I_0(t, \mathbf{x}) = p_L(t)\, \delta_D(r^2) \tag{2.14}$$

Finally, heating power density $\dot{q}(t, \mathbf{x})$ due to a laser pulse with Gaussian beam can be modelled as:

$$\dot{q}(t, \mathbf{x}) = \beta \frac{2p_L(t)}{\pi r_L^2} e^{-2(r/r_L)^2} \tag{2.15}$$

It is clearly seen from Eq. (2.15) that parameter β represents the efficiency of the medium in terms of laser absorption. Then, notice that increasing β and laser power p_L and decreasing characteristic radius r_L lead to higher thermal gradients, and temperature too. It is possible to tune these parameters in order to optimise performance or efficiency of the droplet manipulation system. For instance, overall energy consumption or maximum requested power may be minimized setting larger β and smaller r_L for a given laser power p_L. Differently in biological application it may be advisable to limit temperature variations while moving droplets as fast as possible.

2.3 Temperature field of liquid domain

This Section defines two approaches to model the thermal behaviour of a thin liquid layer when heated by using a laser beam. Heat equation solution procedure is simplified thanks to assumptions that are meaningful for the particular system in analysis. The first approach is based on the fact that heat equation is tractable when heat generation is applied as a series of finite pulses — thanks to assumptions made below and superposition principle. Instead, the second approach relies on spatial discretization via Galerkin finite element method (FEM) and transformation into a (large scale) system of ODEs. Both numerical methods approximate the solution, converging to the actual solution as spatial and time resolution increases — thanks to consistency and stability of FEM and convolution integral approximation.

Thermal field is governed by the heat equation (2.7) and controlled through the heating power density $\dot{q}$, that depends on actual laser position and power. Some assumptions on physical properties of the system are introduced to lower the mathematical complexity of the model. Let us discuss the following assumptions:

1. small temperature variations permit to consider constant homogeneous temperature-independent parameters;

2. mass transport phenomena are negligible in a thin liquid layer [59], that is described by two spatial coordinates;

3. constant homogeneous Dirichlet boundary conditions ensure that temperature along the boundary is uniform and constant[3];

4. spatial distribution of laser intensity is Gaussian, see §2.2.

With these assumptions, projected on a fixed reference frame, given a domain Ω such that $\mathbf{x} \in \Omega$, the model reads

$$\varrho c_p \frac{\partial T}{\partial t}(t, \mathbf{x}) = k \nabla^2_{\mathbf{x}} T(t, \mathbf{x}) + \dot{q}(t, \mathbf{x}) \tag{2.16}$$

and (2.15) describes the heating source.

2.3.1 Semi-analytical solution

A semi-analytical approach to find the thermal field induced by laser heating in a thin liquid layer is here developed. This approach relies on the linearity of heat equation when no heating power is present. Taking advantage of this, temperature field is thought as the convolution of infinitely many instantaneous laser pulses.

Let us denote temperature field w, laser position $\mathbf{x}_L$ and power p_L, heat sink temperature $\bar{w}$, time interval $[0, T]$ and spatial domain Ω with boundary $\partial\Omega$. In this we consider $\mathbf{x}_L \in \mathbb{R}^2$ and laser power $p_L \in [0, +\infty)$.

Unstationary model of an instantaneous axial point-wise heat source

Unsteady heat equation is considered a medium with constant homogeneous physical properties and a heating source with point-wise intensity distribution, centred in $\mathbf{x}_L$, infinitely short duration and overall energy E, namely:

$$I_0(t, \mathbf{x}) = E\, \delta_D(t)\, \delta_D(\|\mathbf{x} - \mathbf{x}_L\|^2) \tag{2.17}$$

The unsteady heat equations holds almost $\forall t \in [0, T]$ and the initial condition is homogeneous, for $\mathbf{x} \in \mathbb{R}^2$:

$$\varrho c_p \frac{\partial w}{\partial t}(t, \mathbf{x}) = k\, \nabla^2_{\mathbf{x}} w(t, \mathbf{x}) + \beta E\, \delta_D(t)\, \delta_D(\|\mathbf{x} - \mathbf{x}_L\|^2) \tag{2.18}$$

$$w(0, \mathbf{x}) = \bar{w} \tag{2.19}$$

Lemma 1. *A solution to Equations (2.18)-(2.19) for $t > 0$ and $\mathbf{x} \in \mathbb{R}^2$ is:*

$$\boxed{w(t, \mathbf{x}) = \bar{w} + \frac{\beta E}{4\pi k} \frac{1}{t} e^{-\|\mathbf{x} - \mathbf{x}_L\|^2/(4\alpha t)}} \tag{2.20}$$

where $\alpha = k/(\varrho c_p)$.

Proof. Dimensional analysis (by using Buckingham's π-theorem) of Eq. (2.18) suggests that temperature field $w(t, \mathbf{x})$ may be a function of the dimensionless coordinate $\eta = r/\sqrt{4\alpha t}$ only, i. e. $w = w(\eta(t, \mathbf{x}))$, where $r = \|\mathbf{x} - \mathbf{x}_L\|$ denotes the distance from

[3] There is no temperature control on the system that keeps a constant temperature, so this assumption is not strictly valid. However, temperature at the boundary can be easily estimated and its variation is expected to be small. This approximation turns out to be useful in both methods exploited for solving the heat equation.

the optical axis. This can be verified, because evaluating partial derivatives in (2.18) for $t > 0$ or $r > 0$ leads to:

$$\begin{gathered} \varrho c_p \frac{\mathrm{d}w}{\mathrm{d}\eta}(\eta) \frac{-r}{2t\sqrt{4\alpha t}} = k \frac{1}{r} \frac{\mathrm{d}}{\mathrm{d}\eta} \left(r \frac{\mathrm{d}w}{\mathrm{d}\eta}(\eta) \frac{1}{\sqrt{4\alpha t}} \right) \frac{1}{\sqrt{4\alpha t}} \\ \Downarrow \\ -\eta^2 \frac{\mathrm{d}w}{\mathrm{d}\eta}(\eta) = \frac{\mathrm{d}}{\mathrm{d}\eta} \left(\eta \frac{\mathrm{d}w}{\mathrm{d}\eta}(\eta) \right) \end{gathered} \tag{2.21}$$

and then a second-order ODE is found:

$$\left(2\eta + \frac{1}{\eta} \right) \frac{\mathrm{d}w}{\mathrm{d}\eta}(\eta) + \frac{\mathrm{d}^2 w}{\mathrm{d}\eta^2}(\eta) = 0 \tag{2.22}$$

A general solution can be found by reducing Eq. (2.22) to a first-order ODE with an auxiliary variable equal to the first derivative w.r.t. η and applying the separation of variables; integrating once again, a solution $w_0(\eta)$ of (2.22) is obtained:

$$w_0(\eta) = c_{w,1} \int \frac{e^{-\eta^2}}{\eta} \mathrm{d}\eta + c_{w,2} \tag{2.23}$$

where scalars $c_{w,1}$ and $c_{w,2}$ are integration constants. Notice that this solution introduces the exponential integral, that has to be evaluated numerically because its integral is not explicitly available. But, recalling that the heat equation is linear w.r.t. time, the time derivative of (2.23), substituted with the expression of η, is another solution of the initial PDE (2.18). It is

$$w(t, \mathbf{x}) = \frac{\partial w_0}{\partial \eta}(\eta(t, \mathbf{x})) \frac{\partial \eta}{\partial t}(t, \mathbf{x}) = -c_{w,1} \frac{1}{2t} e^{-r^2/(4\alpha t)} \tag{2.24}$$

and it is a solution of (2.18), because for $t > 0$ it satisfies:

$$\begin{aligned} \varrho c_p \frac{\partial w}{\partial t}(t, \mathbf{x}) &= k \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial w}{\partial r}(t, \mathbf{x}) \right) \\ &\Downarrow \\ -\frac{c_{w,1}}{2t^2} e^{-\eta^2(t,\mathbf{x})} \left(\eta^2(t, \mathbf{x}) - 1 \right) &= \frac{\alpha}{r} \frac{\partial}{\partial r} \left(c_{w,1} e^{-\eta^2(t,\mathbf{x})} \frac{\eta^2(t, \mathbf{x})}{t} \right) \\ &= c_{w,1} \frac{\alpha}{r} \frac{\partial}{\partial r} \left(e^{-\eta^2(t,\mathbf{x})} \frac{r^2}{4\alpha t^2} \right) \\ &= \frac{c_{w,1}}{2t^2} e^{-\eta^2(t,\mathbf{x})} \left(1 - \eta^2(t, \mathbf{x}) \right) \end{aligned} \tag{2.25}$$

Let us notice that $w(t, \mathbf{x})$ in (2.24) is a solution of (2.18) for $t > 0$ even if a constant term is added. This is needed to satisfy the homogeneous initial condition (2.19), and then it is:

$$w(t, \mathbf{x}) = \bar{w} + \frac{A}{t} e^{-r^2/(4\alpha t)} \tag{2.26}$$

where $A = -c_{w,1}/2$ is a constant scalar to be identified that represents the effect of the laser pulse.

Important properties of this solution have to be discussed, in particular the subsuming assumption of an infinitely large medium. This hypothesis leads to the behaviour at the boundary and at infinite time, recalling that $r = \|\mathbf{x} - \mathbf{x}_L\|$:

$$\lim_{r \to +\infty} w(t, \mathbf{x}) = \bar{w} \qquad \lim_{t \to +\infty} w(t, \mathbf{x}) = \bar{w} \tag{2.27}$$

These properties reflect the assumption of constant homogeneous Dirichlet boundary conditions proposed above, although for $r \to +\infty$.

Constant A in (2.26) rules the effect of laser pulse on the temperature field. It may be computed imposing the energy balance between absorbed energy from the laser beam and the variation of thermal energy inside the (liquid) medium, that has to hold $\forall t > 0$.

$$\begin{aligned} \beta l_z E &= \int_V \varrho c_p (w(t,\mathbf{x}) - \bar{w})\, \mathrm{d}V \\ &= \varrho c_p l_z \int_\Omega \frac{A}{t} e^{-r^2/(4\alpha t)} \mathrm{d}\Omega \\ &= \varrho c_p l_z \frac{A}{t} \lim_{R\to\infty} 2\pi \int_0^R e^{-r^2/(4\alpha t)} r \mathrm{d}r = 4\pi \varrho c_p l_z A \alpha \\ &\Downarrow \\ A &= \frac{\beta E}{4\pi k} \end{aligned} \tag{2.28}$$

Finally, temperature field due to an instantaneous point-wise laser pulse for $t > 0$ is given by (2.26) with $A = \beta E/(4\pi k)$ and Lemma 1 is proved. □

Unstationary model of an instantaneous axial Gaussian heat source

Let us focus on the effect of an infinitely short laser pulse with Gaussian spatial distribution centred in $\mathbf{x}_L$ and overall energy E, namely:

$$I_0(t,\mathbf{x}) = \frac{2E}{\pi r_L^2} \delta_D(t)\, e^{-2(r/r_L)^2} \tag{2.29}$$

The unsteady heat equations holds almost $\forall t \in [0,T]$ and the initial condition is homogeneous, for $\mathbf{x} \in \mathbb{R}^2$:

$$\varrho c_p \frac{\partial w}{\partial t}(t,\mathbf{x}) = k\, \nabla^2_{\mathbf{x}} w(t,\mathbf{x}) + \frac{2\beta E}{\pi r_L^2}\, \delta_D(t)\, e^{-2(r/r_L)^2} \tag{2.30}$$

$$w(0,\mathbf{x}) = \bar{w} \tag{2.31}$$

Lemma 2. *The solution to Equations (2.30)-(2.31) for $t > 0$ and $\mathbf{x} \in \mathbb{R}^2$ is:*

$$\boxed{w(t,\mathbf{x}) = \bar{w} + \frac{\beta E}{\pi \varrho c_p} \frac{2}{8\alpha t + r_L^2} e^{-2\frac{\|\mathbf{x}-\mathbf{x}_L\|^2}{8\alpha t + r_L^2}}} \tag{2.32}$$

where $\alpha = k/(\varrho c_p)$.

Proof. An approach to find a solution of PDE (2.30) for $t > 0$ is matching parameters of a suitable guess function. Let us recall the solution (2.20) for a point-wise laser source and build by similarity the parametrized function $w_p(t,\mathbf{x})$. The idea is to find (if possible) suitable real non-zero scalars a, b and c such that $w_p(t,\mathbf{x})$, defined in (2.33), satisfies (2.30) for $t > 0$. If these parameters can be identified, then existence of a solution is proved; if these parameters are unique, then uniqueness of the solution is proved.

$$w_p(t,\mathbf{x}) = \bar{w} + \frac{a}{ct+1} e^{-b\|\mathbf{x}-\mathbf{x}_L\|^2/(ct+1)} \tag{2.33}$$

Substituting $w_p(t,\mathbf{x})$ into (2.30) and denoting $r = \|\mathbf{x}-\mathbf{x}_L\|$, after some calculations it reads:

$$\frac{a}{(ct+1)^2}\left(\frac{br^2}{ct+1}-1\right)(\varrho c_p c - 4kb)\, e^{-br^2/(ct+1)} = \frac{2\beta E}{\pi r_L^2}\delta_D(t)e^{-2(r/r_L)^2} \tag{2.34}$$

that should hold $\forall t > 0$ and $\forall r \geq 0$. In particular, for $t \neq 0$ heat generation vanishes, and then Eq. (2.34) leads to a relationship between parameters b and c.

$$\begin{gathered}\frac{a}{(ct+1)^2}\left(\frac{br^2}{ct+1}-1\right)(\varrho c_p c - 4kb)\, e^{-br^2/(ct+1)} = 0 \qquad \forall t > 0\,,\ \forall r \geq 0 \\ \Downarrow \\ \varrho c_p c - 4kb = 0 \\ \Downarrow \\ c = 4\alpha b\end{gathered} \tag{2.35}$$

Balance between absorbed energy and variation of internal energy should hold $\forall t > 0$, and hence:

$$\begin{aligned}\beta l_z E &= \int_V \varrho c_p \left(w_p(t,\mathbf{x}) - \bar{w}\right)\,\mathrm{d}V \\ &= \varrho c_p l_z \int_\Omega \frac{a}{ct+1} e^{-br^2/(ct+1)}\,\mathrm{d}\Omega \\ &= \varrho c_p l_z \frac{a}{ct+1} \lim_{R\to+\infty} 2\pi \int_0^R e^{-br^2/(ct+1)} r\,\mathrm{d}r \\ &= \varrho c_p l_z \pi \frac{a}{b} \\ &\Downarrow \\ a &= \frac{\beta E}{\pi \varrho c_p} b\end{aligned} \tag{2.36}$$

At this step, using (2.36) and (2.35), $w_p(t,\mathbf{x})$ can be expressed with parameter b only, namely:

$$w_p(t,\mathbf{x}) = \bar{w} + \frac{\beta E}{\pi \varrho c_p}\frac{b}{4\alpha b t + 1} e^{-br^2/(4\alpha b t+1)} \tag{2.37}$$

Notice that $\forall b \neq 0$ function $w_p(t,\mathbf{x})$ shows the same properties at the boundary of solution (2.20) for the point-wise laser pulse, namely with $r = \|\mathbf{x}-\mathbf{x}_L\|$:

$$\lim_{r\to+\infty} w_p(t,\mathbf{x}) = \bar{w} \qquad\qquad \lim_{t\to+\infty} w_p(t,\mathbf{x}) = \bar{w} \tag{2.38}$$

An additional information is needed to identify a value of parameter b and it is related to the intensity distribution of the laser beam. Let us consider Eq. (2.30) in the spirit of distributions and integrate both sides in the time interval $t \in [0^-, \varepsilon]$ where scalar ε tends to zero.

$$\begin{aligned}\lim_{\varepsilon\to 0^+}\int_{-\varepsilon}^{\varepsilon} \varrho c_p \frac{\partial w}{\partial t}(t,\mathbf{x})\,\mathrm{d}t = \lim_{\varepsilon\to 0^+}\int_{-\varepsilon}^{\varepsilon} k\nabla^2_{\mathbf{x}} w(t,\mathbf{x})\mathrm{d}t + \\ + \lim_{\varepsilon\to 0^+}\int_{-\varepsilon}^{\varepsilon} \frac{2\beta E}{\pi r_L^2} e^{-2(r/r_L)^2}\delta_D(t)\,\mathrm{d}t \\ \Downarrow \\ \lim_{\varepsilon\to 0^+} w(\varepsilon,\mathbf{x}) - \lim_{\varepsilon\to 0^+} w(-\varepsilon,\mathbf{x}) = \frac{2\beta E}{\pi r_L^2 \varrho c_p} e^{-2(r/r_L)^2}\end{aligned} \tag{2.39}$$

Substitution of initial conditions (2.31) shows that a solution has to satisfy the following:

$$\lim_{\varepsilon \to 0^+} w(\varepsilon, \mathbf{x}) = \bar{w} + \frac{2\beta E}{\pi r_L^2 \varrho c_p} e^{-2(r/r_L)^2} \tag{2.40}$$

Let us compute the limit for $t \to 0^+$ of $w_p(t, \mathbf{x})$ given in (2.37), $\mathbf{x} \in \mathbb{R}^2$, and compare the result with Eq. (2.40) to find (if possible) a value of parameter b.

$$\begin{aligned}
\lim_{t \to 0^+} w_p(t, \mathbf{x}) &= \bar{w} + \frac{\beta E}{\pi \varrho c_p} b e^{-br^2} \\
&\Downarrow \\
\bar{w} + \frac{\beta E}{\pi \varrho c_p} b e^{-br^2} &= \bar{w} + \frac{2\beta E}{\pi r_L^2 \varrho c_p} e^{-2(r/r_L)^2} \\
&\Downarrow \\
b e^{-br^2} &= \frac{2}{r_L^2} e^{-2(r/r_L)^2} \\
&\Downarrow \\
b &= \frac{2}{r_L^2}
\end{aligned} \tag{2.41}$$

An unique value has been identified for each parameter in (2.33), satisfying dynamics and initial conditions. Thus, function $w_p(t, \mathbf{x})$ with matched parameters is the unique solution of (2.30)-(2.31) for $t > 0$, and Lemma 2 is proved. □

Collection of pulses

Heat equation (2.16) is the governing equation of the thermal field. It is linear, and thus given the impulse response of the system it is possible to reconstruct its behaviour in response to any input by using the convolution integral and the initial conditions.

Then, the exact thermal field generated by a laser beam trajectory $\mathbf{x}_L(t)$ and power $p_L(t)$ can be computed by the convolution integral. Assuming Gaussian laser beam intensity profile and homogeneous initial condition $w(0, \mathbf{x}) = \bar{w}$, then for $t > 0$ and $\mathbf{x} \in \mathbb{R}^2$ it is:

$$w(t, \mathbf{x}) = \bar{w} + \frac{2}{\pi} \frac{\beta}{\varrho c_p} \int_0^t \frac{p_L(\tau)}{8\alpha(t-\tau) + r_L^2} e^{-2\frac{\|\mathbf{x} - \mathbf{x}_L(\tau)\|^2}{8\alpha(t-\tau) + r_L^2}} \, \mathrm{d}\tau \tag{2.42}$$

and the corresponding temperature gradient is given by:

$$\nabla_{\mathbf{x}} w(t, \mathbf{x}) = -\frac{8}{\pi} \frac{\beta}{\varrho c_p} \int_0^t \frac{p_L(\tau)}{[8\alpha(t-\tau) + r_L^2]^2} e^{-2\frac{\|\mathbf{x} - \mathbf{x}_L(\tau)\|^2}{8\alpha(t-\tau) + r_L^2}} \left(\mathbf{x} - \mathbf{x}_L(\tau)\right)^T \, \mathrm{d}\tau \tag{2.43}$$

The analytical solution of these integrals might be quite complex (if possible). We prefer to compute a numerical approximation, by using standard mid-point quadrature method. A continuous laser beam emitted by a moving device is approximated to a finite time-series of instantaneous laser pulses. Let us build a time grid $\Gamma = \{t_k | t_0 < t_1 < \cdots < t_N\}$ and denote laser position $\mathbf{x}_{L,k} = \mathbf{x}_L(t_k)$ and temperature field approximation $w_k(\mathbf{x}) \approx w(t_k, \mathbf{x})$ for $k = 0, 1, \ldots, N$ and $\mathbf{x} \in \mathbb{R}^2$. Given the homogeneous initial condition $w(0, \mathbf{x}) = \bar{w}$, $\mathbf{x} \in \mathbb{R}^2$, temperature field and its gradient

can be approximated to:

$$w_k(\mathbf{x}) = \bar{w} + \frac{2}{\pi}\frac{\beta}{\varrho c_p}\sum_{j=0}^{k-1}\frac{E_j}{8\alpha(t_k - t_j) + r_L^2}e^{-2\frac{\|\mathbf{x}-\mathbf{x}_{L,j}\|^2}{8\alpha(t_k-t_j)+r_L^2}} \tag{2.44}$$

$$\nabla_{\mathbf{x}} w_k(\mathbf{x}) = -\frac{8}{\pi}\frac{\beta}{\varrho c_p}\sum_{j=0}^{k-1}\frac{E_j}{[8\alpha(t_k - t_j) + r_L^2]^2}e^{-2\frac{\|\mathbf{x}-\mathbf{x}_{L,j}\|^2}{8\alpha(t_k-t_j)+r_L^2}}(\mathbf{x}-\mathbf{x}_{L,j})^T \tag{2.45}$$

where E_j is an estimate of the energy carried by the j-th laser pulse, namely:

$$E_j = (t_{j+1} - t_j)p_{L,j} \approx \int_{t_j}^{t_j+1} p_L(\tau)\,\mathrm{d}\tau \tag{2.46}$$

The maximum time step h in the time grid Γ, i. e. $h = \max_{k=1,\dots,N}(t_k - t_{k-1})$, affects the accuracy of the approximation. In particular, in the limit $h \to 0^+$, finite sums in (2.44)-(2.45) converge to integrals in (2.42)-(2.43), in the spirit of Riemann integral definition. This means that a more refined time grid Γ yields a more accurate approximation of temperature field and temperature gradient.

Steady state model of moving continuous point-wise heat source

Here we derive the temperature field induced by a continuous laser beam moving at constant speed, in the case of point-wise constant intensity distribution and power. This analytical solution is useful to compare results with convolution integral approximations (2.44)-(2.45).

Let us consider w.l.o.g. the heat source moving with at constant non-zero velocity $\mathbf{v}_L$, passing through the origin at time $t = 0$, thus $x_L(t) = \mathbf{v}_L t$. The governing equation (with constant homogeneous parameters and laser power) is given by:

$$\frac{\partial w}{\partial t}(t,\mathbf{x}) = \alpha\nabla_{\mathbf{x}}^2 w(t,\mathbf{x}) + \frac{\beta p_L}{\varrho c_p}\delta_D\left(\|\mathbf{x} - \mathbf{v}_L t\|^2\right) \tag{2.47}$$

Moreover, the following far field conditions have to be satisfied

$$\lim_{t\to\infty} w(t,\mathbf{x}) = \bar{w} \qquad \lim_{\|\mathbf{x}\|\to+\infty} w(t,\mathbf{x}) = \bar{w} \tag{2.48}$$

Lemma 3. *The solution to Equations (2.47)-(2.48) for $\mathbf{x} \in \mathbb{R}^2$ is:*

$$\boxed{w(t,\mathbf{x}) = \bar{w} + \frac{\beta p_L}{2\pi k}\mathcal{K}_0\left(\frac{ur}{2\alpha}\right)e^{-u\xi/(2\alpha)}} \tag{2.49}$$

where $r = \|\boldsymbol{\xi}\|$, $\boldsymbol{\xi} = [\xi,\eta]^T = \mathbf{x} - \mathbf{v}_L t$, $\mathbf{v}_L = [u,0]^T$ and $\mathcal{K}_0$ is the zero-order modified Bessel function of second kind.

Proof. In steady state conditions the temperature field does not change in time from the laser point of view. This means that if the temperature field is described in a reference frame moving with the laser beam, then it does not depend on time. Namely, evaluating the temperature field $\tilde{w}(t,\boldsymbol{\xi}) = w(t,\boldsymbol{\xi} + \mathbf{v}_L t)$, the explicit time dependence disappears, i. e.

$$\begin{aligned} 0 = \frac{\partial \tilde{w}}{\partial t}(t,\boldsymbol{\xi}) &= \frac{\partial w}{\partial t}(t,\boldsymbol{\xi} + \mathbf{v}_L t) + \mathbf{v}_L \cdot \nabla_{\mathbf{x}} w(t,\boldsymbol{\xi} + \mathbf{v}_L t) \\ &\Downarrow \\ -\mathbf{v}_L \cdot \nabla_{\mathbf{x}} w(t,\boldsymbol{\xi} + \mathbf{v}_L t) &= \frac{\partial w}{\partial t}(t,\boldsymbol{\xi} + \mathbf{v}_L t) \end{aligned} \tag{2.50}$$

Noticing that $\mathbf{x} = \boldsymbol{\xi} + \mathbf{v}_L t$ and substituting (2.47) into (2.50) we get:

$$\begin{aligned} -\mathbf{v}_L \cdot \nabla_{\mathbf{x}} w(t, \boldsymbol{\xi} + \mathbf{v}_L t) &= \alpha \nabla^2_{\mathbf{x}} w(t, \boldsymbol{\xi} + \mathbf{v}_L t) + \frac{\beta p_L}{\varrho c_p} \delta_D \left(\|\boldsymbol{\xi} + \mathbf{v}_L t - \mathbf{x}_L(t)\|^2 \right) \\ &\Downarrow \\ -\mathbf{v}_L \cdot \nabla_{\boldsymbol{\xi}} \tilde{w}(t, \boldsymbol{\xi}) &= \alpha \nabla^2_{\boldsymbol{\xi}} \tilde{w}(t, \boldsymbol{\xi}) + \frac{\beta p_L}{\varrho c_p} \delta_D \left(\|\boldsymbol{\xi}\|^2 \right) \end{aligned} \tag{2.51}$$

Then, considering w.l.o.g. the laser moving along one axis of the fixed reference frame, $\mathbf{v}_L = [u, 0]^T$, and coordinates in the moving reference frame $\boldsymbol{\xi} = [\xi, \eta]^T$, then it becomes

$$-u \frac{\partial \tilde{w}}{\partial \xi} = \alpha \nabla^2_{\boldsymbol{\xi}} \tilde{w} + \frac{\beta p_L}{\varrho c_p} \delta_D \left(\|\boldsymbol{\xi}\|^2 \right) \tag{2.52}$$

that is a second-order ODE, hence temperature field $\tilde{w}$ does not depend on time, $\tilde{w} = \tilde{w}(\boldsymbol{\xi})$. Then, far field conditions (2.48) correspond to

$$\lim_{\|\boldsymbol{\xi}\| \to +\infty} \tilde{w}(\boldsymbol{\xi}) = \bar{w} \tag{2.53}$$

Let us consider the scalar function $\varphi(\boldsymbol{\xi}) = (\tilde{w}(\boldsymbol{\xi}) - \bar{w}) e^{u\xi/(2\alpha)}$; it satisfies the following equation:

$$-\alpha \left(\nabla^2_{\boldsymbol{\xi}} \varphi(\boldsymbol{\xi}) - \frac{u^2}{4\alpha^2} \varphi(\boldsymbol{\xi}) \right) e^{-u\xi/2\alpha} = \lambda \delta_D(\|\boldsymbol{\xi}\|^2) \tag{2.54}$$

Denoting $r = \|\boldsymbol{\xi}\|$ the distance from the heating source, Eq. (2.54) becomes:

$$-\alpha \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial \varphi}{\partial r}(\boldsymbol{\xi}) \right) - \left(\frac{u}{2\alpha} \right)^2 \varphi(\boldsymbol{\xi}) \right] e^{-u\xi/2\alpha} = \lambda \, \delta_D(r^2) \tag{2.55}$$

and in the case $r = \|\boldsymbol{\xi}\| > 0$ it reduces to the Bessel equation of order zero, Eq. (2.56). It is an ODE, and hence the solution explicitly depends on coordinate $r = \|\boldsymbol{\xi}\|$ only, Eq. (2.57), thus $\varphi = \varphi(r)$.

$$\frac{1}{r} \frac{\mathrm{d}}{\mathrm{d}r} \left(r \frac{\mathrm{d}\varphi}{\mathrm{d}r}(r) \right) - \left(\frac{u}{2\alpha} \right)^2 \varphi(r) = 0 \tag{2.56}$$

$$\varphi(r) = c_{\varphi,1} \mathcal{I}_0 \left(\frac{ur}{2\alpha} \right) + c_{\varphi,2} \mathcal{K}_0 \left(\frac{ur}{2\alpha} \right) \tag{2.57}$$

Scalars $c_{\varphi,1}$ and $c_{\varphi,2}$ are integration constants and $\mathcal{I}_0$ and $\mathcal{K}_0$ are the zero-order modified Bessel functions of the first and second kinds, respectively.

Far field condition (2.53), denoting $r = \|\boldsymbol{\xi}\|$ and $\xi = r \cos\vartheta \; \forall \vartheta \in [0, \pi]$, leads to:

$$\begin{aligned} 0 &= \lim_{r \to \infty} \left[\tilde{w}(\boldsymbol{\xi}) - \bar{w} \right] \\ &= \lim_{r \to \infty} \varphi(r) \, e^{-ur\cos\vartheta/2\alpha} \\ &= \lim_{r \to \infty} \left[c_{\varphi,1} \mathcal{I}_0 \left(\frac{ur}{2\alpha} \right) + c_{\varphi,2} \mathcal{K}_0 \left(\frac{ur}{2\alpha} \right) \right] e^{-ur\cos\vartheta/2\alpha} \\ &\Downarrow \quad (\sigma = ur/(2\alpha) > 0) \\ 0 &= c_{\varphi,1} \lim_{\sigma \to \infty} \mathcal{I}_0(\sigma) \, e^{-\sigma\cos\vartheta} + c_{\varphi,2} \lim_{\sigma \to \infty} \mathcal{K}_0(\sigma) \, e^{-\sigma\cos\vartheta} \\ &= c_{\varphi,1} \lim_{\sigma \to \infty} \mathcal{I}_0(\sigma) \, e^{-\sigma\cos\vartheta} \\ &\Downarrow \\ 0 &= c_{\varphi,1} \end{aligned} \tag{2.58}$$

Then, the relationship between absorbed power and thermal gradient in laser position (where there is a singularity due to point-wise intensity distribution) returns, $\forall \vartheta \in [0, \pi]$:

$$\begin{aligned}
\beta l_z p_L &= \lim_{r\to 0}\left(-2\pi r l_z k \frac{\partial \tilde{w}}{\partial r}(\boldsymbol{\xi})\right) \\
&= 2\pi l_z k \lim_{r\to 0}\left(-r\frac{\partial}{\partial r}\left(\varphi(r)e^{-ur\cos\vartheta/2\alpha}\right)\right] \\
&\Downarrow \\
\beta p_L &= 2\pi k \lim_{r\to 0}\left[-re^{-ur\cos\vartheta/2\alpha}\left(\frac{\mathrm{d}\varphi}{\mathrm{d}r}(r) - \frac{u}{2\alpha}\varphi(r)\cos\vartheta\right)\right] \\
&\Downarrow \quad (\sigma = ur/(2\alpha) > 0) \\
\beta p_L &= 2\pi k c_{\varphi,2} \lim_{\sigma\to 0}\left[\sigma e^{-\sigma\cos\vartheta}\left(\mathcal{K}_1(\sigma) + \mathcal{K}_0(\sigma)\cos\vartheta\right)\right] = 2\pi k c_{\varphi,2} \\
&\Downarrow \\
c_{\varphi,2} &= \frac{\beta p_L}{2\pi k}
\end{aligned} \tag{2.59}$$

Finally, given $\boldsymbol{\xi} = [\xi, \eta]^T = \mathbf{x} - \mathbf{v}_L t$ and $r = \|\boldsymbol{\xi}\|$, substituting the values of scalars $c_{\varphi,1}$ and $c_{\varphi,2}$ in Eq. (2.57), the Lemma 3 is proved. □

Temperature gradient $\nabla_{\mathbf{x}} w(t, \mathbf{x})$ can be analytically expressed too. A brief comparison between this analytical solution and the approximation through the superposition of a finite series of pulses is presented in Appendix A.

2.3.2 Finite Elements

The finite element method (FEM) is a numerical technique for approximating solutions to boundary value problems (BVPs) for PDEs. The whole domain is subdivided into smaller domains called finite elements (FEs). Governing equations of finite elements and their connections are assembled together; this yields to a set of ODEs. Numerical techniques are used to solve this large set of equations.

Laser-induced temperature field is governed by the unsteady heat equation (2.16), that is a parabolic PDE, thus it is possible to apply the FEM to approximate its evolution. The procedure starts with the discretization of spatial domain, i. e. the generation of nodes and finite elements. The infinite dimensional problem is here reduced into a finite dimensional problem, whose size is the number of nodes (because the thermal field is a scalar field). Then, shape functions interpolate both the main variable and the weight variable (called also virtual variable) in the finite element domain. This means that it is possible to approximate thermal and virtual field inside each element as a function of space and nodal values, i. e. the values of field at the nodes. Then, the problem is converted into its weak formulation for each element, equations are assembled into mass and stiffness matrix and finally the problem is solved with standard tools of linear algebra. For a deeper and more detailed discussion about FEM see [60].

In the framework of FE, the evaluation of thermal gradient $\nabla_{\mathbf{x}} w(t, \mathbf{x})$ is approximated by a matrix multiplication:

$$\nabla_{\mathbf{x}} w(t, \mathbf{x})^T \approx \boldsymbol{B}(\mathbf{x})\mathbf{w}(t) \tag{2.60}$$

where nodal vector $\mathbf{w} = \mathbf{w}(t)$ depends on time and matrix $\boldsymbol{B}$ is computed from shape functions $\mathbf{n}$ and depends on spatial position $\mathbf{x}$ (where the gradient has to be evaluated).

In particular, it is $\boldsymbol{B}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathbf{n}(\mathbf{x})$. Thermal field dynamics is represented by a system of ODEs, that is worth to express in matrix form as:

$$\boxed{\boldsymbol{M}_w\dot{\mathbf{w}}(t) + \boldsymbol{K}_w\mathbf{w}(t) = \mathbf{f}(t)} \tag{2.61}$$

where $\boldsymbol{M}_w$ and $\boldsymbol{K}_w$ are mass and stiffness matrix, respectively. Vector of nodal loads $\mathbf{f}$ corresponds to the heat generation term in (2.16). The structure of mass matrix $\boldsymbol{M}_w$ and stiffness matrix $\boldsymbol{K}_w$ depends on node numbering and values of their entries are influenced by element sizes and physical properties of the liquid layer. Considering constant homogeneous temperature-independent properties, these matrices are proportional to physical properties of the liquid. The same holds for the nodal load vector $\mathbf{f}$ w.r.t. absorption coefficient.

$$\boldsymbol{M}_w = l_z\varrho c_p\bar{\boldsymbol{M}}_w\ , \qquad \boldsymbol{K}_w = l_z k\bar{\boldsymbol{K}}_w\ , \qquad \mathbf{f} = \beta\bar{\mathbf{f}} \tag{2.62}$$

Nodal loads Let us focus on how to compute the vector of nodal loads $\mathbf{f}$. In general, the nodal load vector of element e depends on shape functions $\mathbf{n}_e(\mathbf{x})$ and control input; namely at time t:

$$\mathbf{f}_e(t) = \int_{\Omega_e} \mathbf{n}_e(\mathbf{x})\,\dot{q}(t,\mathbf{x})\mathrm{d}\Omega_e \tag{2.63}$$

where domain Ω_e is the e-th element. Scalar function $\dot{q} = \dot{q}(t,\mathbf{x})$ is the heating power per unit volume and may depend on both time and space, as explained in §2.2. Considering a point-wise laser source (equivalent to a Gaussian one with null characteristic radius, and thus modelled with a Dirac delta distribution) placed in $\mathbf{x}_L$ with power p_L, it is

$$\begin{aligned}\mathbf{f}_e &= \int_{\Omega_e} \mathbf{n}_e(\mathbf{x})\delta_D(\|\mathbf{x}-\mathbf{x}_L\|)\beta p_L\mathrm{d}\Omega_e \\ &= \beta p_L\mathbf{n}_e(\mathbf{x}_L)\end{aligned} \tag{2.64}$$

Instead, in the case of a (proper) Gaussian intensity distribution of the laser beam, it is

$$\begin{aligned}\mathbf{f}_e &= \int_{\Omega_e} \mathbf{n}_e(\mathbf{x})\frac{2}{\pi r_L^2}e^{-2(\|\mathbf{x}-\mathbf{x}_L\|/r_L)^2}\beta p_L\mathrm{d}\Omega_e \\ &= \frac{2}{\pi r_L^2}\beta p_L\int_{\Omega_e} \mathbf{n}_e(\mathbf{x})e^{-2(\|\mathbf{x}-\mathbf{x}_L\|/r_L)^2}\mathrm{d}\Omega_e\end{aligned} \tag{2.65}$$

Integral in (2.65) is quite involved and it is difficult, perhaps impossible, to solve it analytically. Its solution may be approximated through numerical methods, e. g. Gauss quadrature method for integrating generic functions.

Here we propose and implement a procedure inspired by Gaussian quadrature rule and that satisfies the energy conservation law. This procedure is a partition of laser power on grid nodes, weighted on the specific laser intensity distribution. In particular, given a set of nodes $\mathbf{x}_i$ for $i = 1,\dots,n$, we use intensity distribution $\psi(t,\mathbf{x})$—whatever it is—as a weighting function, i. e. to compute the relative weight $w_i(t)$ of node $\mathbf{x}_i$, Eq. (2.66), and then each entry $f_i(t)$ of nodal load vector $\mathbf{f}(t)$ is a weighted partition of the overall heating power $\beta p_L(t)$, Eq. (2.67), where $p_L(t) = \int_{\mathbb{R}^2}\psi(t,\mathbf{x})\,\mathrm{d}\Omega$.

$$w_i(t) = \psi(t,\mathbf{x}_i) \tag{2.66}$$

$$f_i(t) = \beta p_L(t)\frac{w_i(t)}{\sum_{i=1}^n w_i(t)} \tag{2.67}$$

Conservation of energy is guaranteed[4] because $\sum_{i=1}^{n} f_i(t) = \beta p_L(t)$, meaning that the total nodal load is the heating power. This method interestingly resembles 1-point Gauss quadrature approximation of integral in (2.65), sampling function value only in position $\mathbf{x}_i$. Thus, this approach likely works better with smooth functions and homogeneous fine space discretization, where function $\psi(t, \mathbf{x})$ may be considered almost piece-wise constant in each finite element.

Time integration Let us focus now on how to integrate the time evolution of the thermal field given a control history, i. e. given laser position $\mathbf{x}_L(t)$ and power $p_L(t)$ at time t. As previously explained, nodal load vector $\mathbf{f} = \mathbf{f}(t)$ can be computed and expressed as a function of time only.

Considering time-varying Dirichlet boundary conditions[5], nodal temperature vector $\mathbf{w}$ may be partitioned into free and constrained nodes, $\mathbf{w}_\mathrm{f}$ and $\mathbf{w}_\mathrm{c}$ respectively, and values of solution at constrained nodes is given by boundary conditions, i. e. $\mathbf{w}_\mathrm{c}(t) = \mathbf{w}_\mathrm{c}^*(t)$. Thermal dynamics in (2.61) can be written as

$$\begin{bmatrix} \boldsymbol{M}_\mathrm{ff} & \boldsymbol{M}_\mathrm{fc} \\ \boldsymbol{M}_\mathrm{cf} & \boldsymbol{M}_\mathrm{cc} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{w}}_f(t) \\ \dot{\mathbf{w}}_c^*(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_\mathrm{ff} & \boldsymbol{K}_\mathrm{fc} \\ \boldsymbol{K}_\mathrm{cf} & \boldsymbol{K}_\mathrm{cc} \end{bmatrix} \begin{bmatrix} \mathbf{w}_\mathrm{f}(t) \\ \mathbf{w}_\mathrm{c}^*(t) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_f(t) \\ \mathbf{f}_c(t) \end{bmatrix} \tag{2.68}$$

where also matrices $\boldsymbol{M}_w$ and $\boldsymbol{K}_w$ and nodal load vector $\mathbf{f}$ have been partitioned accordingly. Exploiting the knowledge of boundary conditions, vector $\mathbf{w}_\mathrm{f}$ is the only unknown, and considering only dynamics of free nodes system may be reduced. Notice that the r.h.s. term $\boldsymbol{\varphi}_\mathrm{f}$ still depends only on time, Eq. (2.69).

$$\boldsymbol{M}_\mathrm{ff}\dot{\mathbf{w}}_f(t) + \boldsymbol{K}_\mathrm{ff}\mathbf{w}_\mathrm{f}(t) = \mathbf{f}_f(t) - \boldsymbol{M}_\mathrm{fc}\dot{\mathbf{w}}_c^*(t) - \boldsymbol{K}_\mathrm{fc}\mathbf{w}_\mathrm{c}^*(t) = \boldsymbol{\varphi}_\mathrm{f}(t) \tag{2.69}$$

Let us discretize (2.69) obtaining a finite dimensional problem, over a time grid $\Gamma = \{0 = t_0 < t_1 < \cdots < t_N = T\}$, and denote $\Delta t_{k+1} = t_{k+1} - t_k$ the time step size. Equations 2.69 are approximated by using the trapezoidal quadrature rule to average on $[t_k, t_{k+1}]$ and by using finite differences in place of the derivative terms:

$$\begin{gathered} \frac{1}{\Delta t_{k+1}} \int_{t_k}^{t_{k+1}} \left[\boldsymbol{M}_\mathrm{ff}\dot{\mathbf{w}}_f(t) + \boldsymbol{K}_\mathrm{ff}\mathbf{w}_\mathrm{f}(t)\right] dt = \frac{1}{\Delta t_{k+1}} \int_{t_k}^{t_{k+1}} \boldsymbol{\varphi}_\mathrm{f}(t) dt \\ \Downarrow \\ \boldsymbol{M}_\mathrm{ff} \frac{\mathbf{w}_\mathrm{f}(t_{k+1}) - \mathbf{w}_\mathrm{f}(t_k)}{\Delta t_{k+1}} + \boldsymbol{K}_\mathrm{ff} \frac{\mathbf{w}_\mathrm{f}(t_{k+1}) + \mathbf{w}_\mathrm{f}(t_k)}{2} = \frac{\boldsymbol{\varphi}_\mathrm{f}(t_{k+1}) + \boldsymbol{\varphi}_\mathrm{f}(t_k)}{2} + \mathcal{O}(\Delta t_{k+1}^2) \end{gathered} \tag{2.70}$$

Neglecting the truncation term of order $\mathcal{O}(\Delta t_{k+1}^2)$ we obtain the discrete-time update equation:

$$\begin{gathered} \boldsymbol{M}_\mathrm{ff} \frac{\mathbf{w}_{f,k+1} - \mathbf{w}_{f,k}}{\Delta t_{k+1}} + \boldsymbol{K}_\mathrm{ff} \frac{\mathbf{w}_{f,k+1} + \mathbf{w}_{f,k}}{2} = \frac{\boldsymbol{\varphi}_{\mathrm{f},k+1} + \boldsymbol{\varphi}_{\mathrm{f},k}}{2} \\ \Downarrow \\ \left(\boldsymbol{M}_\mathrm{ff} + \frac{\Delta t_{k+1}}{2}\boldsymbol{K}_\mathrm{ff}\right) \mathbf{w}_{f,k+1} = \left(\boldsymbol{M}_\mathrm{ff} - \frac{\Delta t_{k+1}}{2}\boldsymbol{K}_\mathrm{ff}\right) \mathbf{w}_{f,k} + \frac{\Delta t_{k+1}}{2} \left(\boldsymbol{\varphi}_{\mathrm{f},k+1} + \boldsymbol{\varphi}_{\mathrm{f},k}\right) \end{gathered} \tag{2.71}$$

[4] Actually this is true only if rounding errors are neglected. However, energy balance obtained by using Gaussian quadrature could be less accurate.

[5] This is a type of boundary condition imposed on an ordinary or a partial differential equation. In particular, it specifies the values that a solution takes at the boundary of the domain.

where $\mathbf{w}_k \approx \mathbf{w}(t_k)$ denotes the state approximation. Finally, notice that this scheme is explicit because $\boldsymbol{\varphi}_{\mathrm{f},k+1}$ depends on boundary conditions, laser heating and not on the actual thermal field.

2.4 Droplet and thermocapillary force modelling

Liquid medium and droplet motion are governed by principles of fluid mechanics, that are momentum and energy balances; written in differential form, these are the Navier-Stokes equations. These relate changes of the velocity field to the stresses in the fluid, and this detailed approach often deals with involved differential calculus and typically is covered in literature assuming small Reynolds number and Marangoni number[6], in [59, 61, 62] for instance.

An higher-level model may be formulated using the Newton's approach, i. e. considering the droplet as a point mass and summing all the forces applied on it. With this approach, a simpler dynamical model may be formulated, based on an averaged dynamics.

Steady-state migration model and velocity is known from [59]. Unsteady droplet migration has been studied with the aforementioned assumptions with Navier-Stokes equations [61]. Let us estimate the Reynolds number Re to evaluate whether a steady-state model of migration may be enough or not. Reference parameters may be the liquid specific mass $\varrho = 1000\,\mathrm{kg/m^3}$, the characteristic velocity $v = 2 - 5\,\mathrm{mm/s}$, the characteristic length $L = 2 - 4\,\mathrm{mm}$ and the liquid viscosity $\mu = 0.8\,\mathrm{mPa\,s}$, yielding to

$$Re = \frac{\varrho v L}{\mu} \approx 5 - 30 \tag{2.72}$$

Thus, liquid layer exhibits a laminar flow, that is in between creeping flow and inviscid flow, i. e. both inertia and viscous forces are significant, and so a transient model should be elaborated.

Unsteady dynamics modelling is straightforward using Newton's approach, and in particular the Newton's second law of motion, that describes the relationship between a body and the forces acting upon it. Thus, identification of these forces is the key step in modelling the dynamics of a system.

Assumptions In this work we restrict to consider a small lensed[7] droplet, floating on a liquid layer; liquid and droplet are immiscible. Droplet shape deformation and internal convective flow can be omitted is droplet size is small enough. Similarly, liquid temperature field may be imposed to be valid in the droplet too. For this reason, exploting Newton's approach and considering the droplet as a point mass, we will refer to droplet temperature as the temperature of liquid in droplet (central) position; this imply that the same holds for the temperature gradient. The liquid layer is supposed to be homogeneous and thin, so that convective flow can be neglected [59] and variables are constant along the liquid thickness. Hence the temperature field is 2-dimensional.

Physical properties of liquid medium and droplet depend on the environment, and particularly on the surrounding temperature field. These parameters may be

[6] Reynolds number $Re = \varrho v L/\mu$ is the dimensionless ratio of inertial forces to viscous forces; Marangoni number $Ma = -L\sigma_T \Delta T/(\mu\alpha)$ is the dimensionless ratio between thermal-surface tension forces to viscous forces (specific mass ϱ, speed v, length L, viscosity μ, temperature T, surface tension coefficient $\sigma_T = \frac{\partial \sigma}{\partial T}$, thermal diffusivity α).

[7] It has been experimentally highlighted that lensed droplets do not behave as spherical droplets [8].

considered as simple functions of temperature, approximating the non-linear behaviour with piece-wise constant, linear or more complex models. Temperature-dependence of model parameters may yield to a non-linear model of thermocapillary droplet migration [63]. For the sake of simplicity, we will consider constant and homogeneous physical properties of both the liquid medium and the droplet.

Drag force due to surrounding liquid turns out to be linear by using (linear) Navier-Stokes equations with Stokes' approximation (i. e. $Re \ll 1$) [59, 61]. Recent experimental evidences report a good fitting with quadratic drag force [54]. In the following we analyse both the linear and quadratic drag force.

Thermocapillary force Surface tension acts on the free surface of the fluid and hence hydrodynamic stresses balance with the surface tension and its gradient [17]. Thus, thermocapillary force acting on the liquid depends linearly on the surface tension gradient.

Surface tension of a liquid may depend on its chemical composition and its surrounding environment, e. g. electrical, magnetic and temperature field. Surface tension as a function of temperature may be predicted using the Eötvös rule [64], that suggests a linear relation for temperatures below the critical point of the liquid. Thus, thermocapillary force acting on the liquid depends linearly on the temperature gradient. Notice that for most of the liquids surface tension decreases by increasing the temperature [17].

Motion of the liquid medium due to temperature gradients is from warmer regions, with the lower surface tension, toward the colder regions, with the higher surface tension [8, 59, 62]. Classical works suggest that reactive force of surrounding medium would cause a droplet movement opposite to liquid medium flow at the free surface, thus in the direction of the temperature gradient [59, 62]. More recent developments show that lens-shaped droplets at the free surface of immiscible liquids move in the direction of the outward liquid flow, i. e. opposite to the temperature gradient [8]. On the other hand, spherical-shaped droplets move from cooler to warmer regions (because of the depression on the free surface in the hottest region) [8, 13]. However, reactive or not, the force applied onto the droplet depends on the surrounding medium. We conclude that thermocapillary force on the droplet depends linearly on temperature gradient, so as the force on the liquid medium.

Steady-state migration From classical works (with viscous incompressible liquid, Stokes' approximation, no convective flow, constant temperature gradient) [59, 62], the steady state thermocapillary migration velocity of a droplet on a thin liquid layer subject to a constant temperature gradient $\mathbf{g}_{w,\infty}$ is:

$$\mathbf{v}_\infty = -\frac{2R\sigma_T}{\left(2+\frac{\alpha'}{\alpha}\right)\left(2+3\frac{\mu'}{\mu}\right)\mu}\mathbf{g}_{w,\infty} \tag{2.73}$$

where μ is the dynamic viscosity of the liquid and α its thermal diffusivity, R is the droplet radius[8] and $\sigma_T = \frac{\partial\sigma}{\partial w}$ the surface tension coefficient. Terms with prime relate to droplet properties, without to liquid properties.

Linear drag Dill & Balasubramaniam [61] apply Navier-Stokes equations (with small Reynolds number and Marangoni number); their result is a transfer function

[8] Lens-shaped droplet is approximated to an ellipsoid with two equal axis. In particular, droplet thickness along the liquid layer is smaller than the other (almost equal) sizes. We will consider droplet volume V and top-view radius R.

between droplet velocity and thermal gradient. Let us consider the equation of motion $m\dot{\mathbf{v}} = \sum_i \mathbf{f}_i$, where m is the droplet mass and $\mathbf{v} = \dot{\mathbf{x}}$ its velocity, and forces due to surface tension and viscosity. Denoting the temperature gradient $\mathbf{g}_w(t, \mathbf{x}) = \nabla_{\mathbf{x}} w(t, \mathbf{x})$, the equation of motion reads:

$$m\dot{\mathbf{v}}(t) = \mathbf{f}_\sigma(t) + \mathbf{f}_{\text{drag}}(t) = c_1 R^2 \, \sigma_T \mathbf{g}_w(t, \mathbf{x}(t)) - 3\pi\mu R \, \mathbf{v}(t) \tag{2.74}$$

A linear model for thermocapillary force is adopted because of the reasons explained above; linear drag force (derived from Stokes' approximation) is used, to gain a linear model. Thus, it is possible to apply Laplace transform and compare with the unsteady model obtained in [61]. Notice that parameter c_1 is left free for subsequent fitting or comparisons; drag force is modelled with a factor 3π instead of 6π because the droplet is not completely immersed. Actually, droplets are typically not spherical and not half immersed [8, 65].

Looking at the steady state velocity $\mathbf{v}_\infty$ in a constant temperature gradient $\mathbf{g}_{w,\infty}$, from (2.74) it is:

$$\mathbf{v}_\infty = \frac{c_1 R \, \sigma_T}{3\pi\mu} \mathbf{g}_{w,\infty} \tag{2.75}$$

This value may be compared with the steady-state velocity (2.73) from [62]; the two results are equivalent with the particular choice of parameter c_1:

$$c_1 = -\frac{6\pi}{\left(2 + \frac{\alpha'}{\alpha}\right)\left(2 + 3\frac{\mu'}{\mu}\right)} \tag{2.76}$$

Assuming similar physical properties of liquid and droplet, i. e. $\mu' \approx \mu$ and $\alpha' \approx \alpha$, the parameter reduces to $c_1 = -2\pi/5$.

Let us focus on the time constant of the first-order model just introduced. Rearranging the equation of motion in the form $\tau\dot{\mathbf{v}} + \mathbf{v} = \mathbf{v}_\infty$, it reads:

$$\frac{m}{3\pi\mu R}\dot{\mathbf{v}} + \mathbf{v} = \frac{c_1 R \, \sigma_T}{3\pi\mu} \mathbf{g}_{w,\infty} \tag{2.77}$$

where the time constant is highlighted. Let us estimate its value using reference parameters, droplet volume $V = 20 - 30\,\mu\text{L}$ and radius $R = 2 - 3\,\text{mm}$; it is:

$$\tau = \frac{m}{3\pi\mu R} = \frac{V\varrho}{3\pi\mu R} \approx 0.9 - 2.0\,\text{s} \tag{2.78}$$

For comparison, time constant of thermal dynamics in liquid medium is about $\tau_\text{T} = L^2/\alpha \approx 28 - 112\,\text{s}$, for reference length $L = 2 - 4\,\text{mm}$ and thermal diffusivity of water $\alpha = 0.143\,\text{mm}^2/\text{s}$. Hence, from this viewpoint, unsteady migration of droplet could be neglected, being $\tau \ll \tau_\text{T}$.

Let us evaluate the braking distance of the droplet, namely the distance d travelled to stop from the steady state velocity in a time interval equal to one time constant τ. Actually, notice that this is an underestimation of the needed space to stop. Steady state velocity can be estimated from (2.73) with $\alpha' = \alpha$ and $\mu' = \mu$; for $R = 3\,\text{mm}$, $\sigma_T = -0.15\,\text{mN/m K}$ and $\|\mathbf{g}_{w,\infty}\| = 10 - 100\,\text{K/m}$, it is $\|\mathbf{v}_\infty\| \approx 0.75 - 7.5\,\text{mm/s}$. These values match with experimental evidences [8, 54].

Considering uniform deceleration $\mathbf{a} = -\mathbf{v}_\infty/\tau$ and $\tau \approx 0.9 - 2.0\,\text{s}$, braking distance turns out to be (greater than) $d = \|\mathbf{v}_\infty\|\tau/2 \approx 0.3 - 7\,\text{mm}$. Thus, depending on the manipulation and positioning accuracy required by the specific application, the unsteady dynamics may be ignored or not. In this work we consider a transient model for droplet thermocapillary migration.

Let us now compare model (2.74) with results for bubbles in a constant temperature gradient, by using Navier-Stokes equations, obtained in [61]. To this end and without loss of generality, let us consider the 1-dimensional case. Defining dimensionless time $\hat{t} = t\mu/(R^2\varrho)$ and velocity $\hat{v} = v\mu/(\sigma_T R\, g_{w,\infty})$, they found the Laplace transform of the bubble velocity $\hat{v}(\hat{s})$ to be

$$\hat{v}(\hat{s}) = \frac{\frac{1}{\hat{s}}\left(1 + \frac{1+\sqrt{\hat{s}Pr}}{2+2\sqrt{\hat{s}Pr}+\hat{s}Pr}\right)}{3 + \frac{\hat{s}(3+\sqrt{\hat{s}})}{6(1+\sqrt{\hat{s}})}} \qquad \text{(DB)} \tag{2.79}$$

where Pr is the dimensionless Prandtl number, defined as the ratio of momentum diffusivity to thermal diffusivity, whose value is $Pr = \mu c_p/k \approx 5.6$ (specific heat $c_p = 4186\,\mathrm{J/kg\,K}$ and thermal conductivity $k = 0.6\,\mathrm{W/m\,K}$ of water). We can express Eq. (2.74) with dimensionless variables $\hat{t}$ and $\hat{v}$, substituting $m = V\varrho'$ and $V = \frac{4}{3}\pi\varepsilon R^3$ (factor ε accounts for droplet lensed-shape)[9] and considering $\varrho' \approx \varrho$.

$$\begin{aligned} m\dot{v}(t) &= c_1 R^2 \sigma_T \mathbf{g}_{w,\infty} - 3\pi\mu R v(t) \\ &\Downarrow \\ \frac{V}{R^3}\hat{v}'(\hat{t}) &= c_1 - 3\pi\hat{v}(\hat{t}) \\ &\Downarrow \\ \frac{4\varepsilon}{9}\hat{v}'(\hat{t}) &= \frac{c_1}{3\pi} - \hat{v}(\hat{t}) \end{aligned} \tag{2.80}$$

Then we project into the frequency domain by using the Laplace transform, considering null initial conditions, $\hat{v}(t)|_{t=0} = 0$.

$$\begin{aligned} \frac{4\varepsilon}{9}\hat{s}\,\hat{v}(\hat{s}) &= \frac{c_1}{3\pi}\frac{1}{\hat{s}} - \hat{v}(\hat{s}) \\ &\Downarrow \\ \hat{v}(\hat{s}) &= \frac{c_1}{3\pi}\left(\frac{1}{\hat{s}} - \frac{1}{\hat{s}+\frac{9}{4\varepsilon}}\right) \qquad \text{(FO)} \end{aligned} \tag{2.81}$$

In order to compare predicted steady-state velocity $v_\infty = \lim_{t\to\infty} v(t)$, we can apply the final value theorem on both models (2.79) and (2.81). Similarly, initial value theorem can be applied twice to match initial acceleration $\dot{v}(t)|_{t=0}$[10].

$$\lim_{\hat{t}\to\infty} \hat{v}(\hat{t}) = \lim_{\hat{s}\to 0} \hat{v}(\hat{s})\,\hat{s} = \begin{cases} \frac{1}{2} & \text{(DB)} \\ \frac{c_1}{3\pi} & \text{(FO)} \end{cases} \quad \rightarrow \; c_1 = \frac{3}{2}\pi \tag{2.82}$$

$$\lim_{\hat{t}\to 0} \hat{v}'(\hat{t}) = \lim_{\hat{s}\to\infty} \hat{v}(\hat{s})\,\hat{s}^2 = \begin{cases} 6 & \text{(DB)} \\ \frac{3c_1}{4\pi\varepsilon} & \text{(FO)} \end{cases} \quad \rightarrow \; c_1 = 8\pi\varepsilon \tag{2.83}$$

The proposed first-order linear model (2.81) can be compared with (2.79) in the Laplace domain, matching steady state velocity v_∞ or initial acceleration $a(0)$. Fig. 2.3 shows the good agreement between these models, especially for $\varepsilon \approx 0.3$.

[9] Actually droplet shape is not ellipsoidal, as shown in [65].

[10] Null initial velocity $v(t)|_{t=0}$ is set by initial conditions in both models, (2.79) and (2.81). Actually, applying initial value theorem, it is $\lim_{\hat{t}\to 0} \hat{v}(\hat{t}) = \lim_{\hat{s}\to\infty} \hat{v}(\hat{s})\hat{s} = 0$.

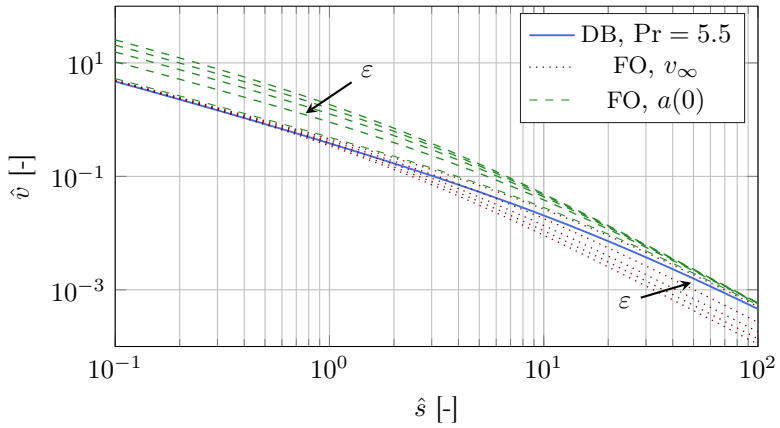


Figure 2.3: Comparison between linear models DB and FO in Laplace domain: effect of shape factor $\varepsilon \in [0.2, 1]$.

Finally, a second-order linear model has been formulated for unsteady thermocapillary migration of a small droplet on a thin liquid layer. Changes in droplet and liquid properties may be embedded considering a linear time-varying model. Condensing parameters in (possibly) time-varying scalars m_D, b_D and c_D to be identified, the aforementioned model reads:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{v}(t) \\ m_D \dot{\mathbf{v}}(t) &= b_D \mathbf{g}_w(t, \mathbf{x}(t)) - c_D \mathbf{v}(t) \end{aligned} \tag{2.84}$$

Quadratic drag Experimental work on motion of floating objects successfully fitted tracking data using a quadratic drag model [8,54]. Assuming that droplet-liquid relative velocity is similar to the droplet absolute velocity[11], the quadratic drag force can be modelled introducing drag coefficient c_x and frontal area A of the droplet, namely:

$$\mathbf{f}_{\text{drag}} = -\frac{1}{2} \varrho c_x A \|\mathbf{v}\| \mathbf{v} \tag{2.85}$$

Coefficient c_x quantifies the resistance of an object in a fluid environment; it depends on its shape and typically it is scaled to specific mass of the fluid and frontal area of the object. It approximately ranges in $c_x \approx 0.04 - 0.4$ for smooth ellipsoidal shapes. Frontal area can be approximated to $A = \pi \varepsilon R^2$, with shape factor ε defined above.

From the second Newton's law we get the (non-linear) model, by considering linear thermocapillary force and quadratic drag:

$$m \dot{\mathbf{v}}(t) = c_1 R^2 \sigma_T \mathbf{g}_w(t, \mathbf{x}(t)) - \frac{1}{2} \varrho c_x A \|\mathbf{v}(t)\| \mathbf{v}(t) \tag{2.86}$$

Compared to (2.75), steady-state velocity $\mathbf{v}_\infty$ is no more proportional to the constant temperature gradient $\mathbf{g}_{w,\infty}$, and interestingly depends on specific mass ϱ rather than viscosity μ of the surrounding liquid.

$$\mathbf{v}_\infty = \left(\frac{4 c_1 \sigma_T}{\pi \varepsilon \varrho c_x} \|\mathbf{g}_{w,\infty}\| \right)^{1/2} \frac{\mathbf{g}_{w,\infty}}{\|\mathbf{g}_{w,\infty}\|} \tag{2.87}$$

[11] Actually, this may be not valid because the droplet moves with the liquid flow [8]. Thus, drag is expect to be small because of the slow relative motion.

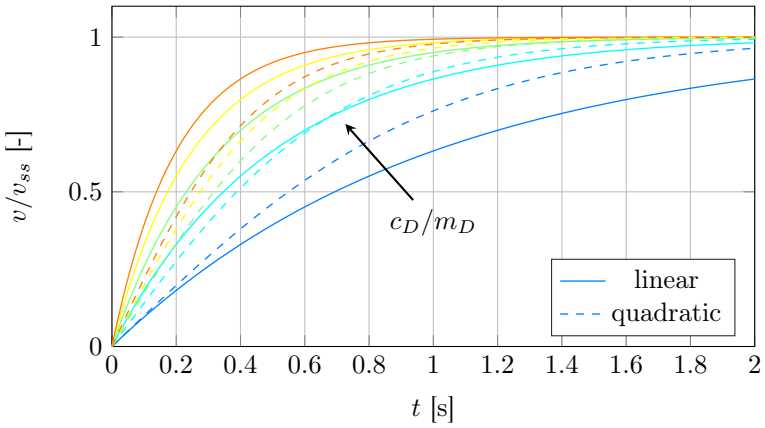


Figure 2.4: Comparison of linear and quadratic model: normalized velocity profile for different values of c_D/m_D ($b_D\|\mathbf{g}_w\|/m_D = 1\,\mathrm{N/kg}$).

A second-order non-linear model has been formulated for unsteady thermocapillary migration of a small droplet on a thin liquid layer. With parameters m_D, b_D and c_D to be identified, this model reads:

$$\boxed{\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{v}(t)\\ m_D\dot{\mathbf{v}}(t) &= b_D\mathbf{g}_w(t,\mathbf{x}(t)) - c_D\|\mathbf{v}(t)\|\mathbf{v}(t)\end{aligned}} \tag{2.88}$$

Summary Let us express the dynamical model with linear thermocapillary force and quadratic drag as a first-order system of ODEs in matrix form. Actually, the model is non-linear, but we can think of an equivalent time-varying linear system. Collecting droplet state vector $\mathbf{z} = [\mathbf{z}^x, \mathbf{z}^v]^T = [\mathbf{x}, \mathbf{v}]^T$, Eq. (2.88) can be written as

$$\boldsymbol{M}_z\dot{\mathbf{z}}(t) = \boldsymbol{A}_z(\mathbf{z}(t))\,\mathbf{z}(t) + \boldsymbol{B}_z\,\mathbf{g}_w(t,\mathbf{z}^x(t)) \tag{2.89}$$

where matrices $\boldsymbol{M}_z$, $\boldsymbol{A}_z$ and $\boldsymbol{B}_z$ depend on properties of droplet and surrounding liquid and droplet state:

$$\boldsymbol{M}_z = \begin{bmatrix}\boldsymbol{I} & 0\\ 0 & m_D\boldsymbol{I}\end{bmatrix}, \qquad \boldsymbol{A}_z(\mathbf{z}) = \begin{bmatrix}0 & \boldsymbol{I}\\ 0 & -c_D\|\mathbf{z}^v\|\boldsymbol{I}\end{bmatrix}, \qquad \boldsymbol{B}_z = \begin{bmatrix}0\\ b_D\boldsymbol{I}\end{bmatrix} \tag{2.90}$$

Temperature gradient could be computed by using a FE model of the liquid medium, as proposed in §2.3.2. Being $\mathbf{w}$ the corresponding nodal temperature vector, it is $\mathbf{g}_w(t,\mathbf{z}) \approx \boldsymbol{B}(\mathbf{z})\mathbf{w}(t)$, where matrix $\boldsymbol{B}$ is a suitable linear operator based on shape functions adopted for the FE discretization, §2.3.2. Hence, re-defining input matrix $\boldsymbol{B}_z(\mathbf{z})$, model of droplet dynamics (2.92) is found.

$$\boldsymbol{B}_z(\mathbf{z}) = \begin{bmatrix}0\\ b_D\boldsymbol{B}(\mathbf{z})\end{bmatrix} \tag{2.91}$$

the model reads

$$\boldsymbol{M}_z\dot{\mathbf{z}}(t) = \boldsymbol{A}_z(\mathbf{z}(t))\,\mathbf{z}(t) + \boldsymbol{B}_z(\mathbf{z}(t))\,\mathbf{w}(t) \tag{2.92}$$

Chapter 3

State and parameters estimation

This Chapter discusses about state estimation and parameters identification of the physical system that comprises droplet, liquid layer, laser and robot. This is a key step that enables us to validate the mathematical model built in Chapter 2 and then to use it in a model-based control strategy, exploited in Chapter 4.

State estimation provides an estimate of the actual internal state of the system, by using measurements of the input and/or output of the real system and its model. Typically, the physical state of the system cannot be fully determined by direct observation and so its mathematical model is used to reconstruct and estimate the hidden part of the state. Similarly, parameters identification refers to the estimation of values of model parameters that best describe the physical system. In principle, we could consider parameters as states with a constant (or slow) dynamics.

State estimation and parameters identification are fundamental to build an effective automatic control system. Actually, a control system based on a feedback loop elaborates and computes a control action to be applied on the system starting from the knowledge of the actual state. Thus, a suitable state estimate is needed. Then, a model-based controller uses an internal representation, i. e. a model, of the physical system to compute a control action. Model parameters affect how the system behaves and evolves, and so an estimate of actual system parameters is required. Notice that the mathematical model is given and not estimated; this means that we trust in it but we may validate the model through experiments, while identifying parameters for instance, verifying that model accurately predicts behaviours of the real system.

Parameters identification may be performed only once during a calibration phase (offline) or more than once while system is evolving (online). The latter approach may be more complicated but likely it yields better prediction of system behaviour[1]. Continuous tracking takes into account slowly time-varying physical effects on the system, e. g. temperature-dependence of properties, and also the set of parameters that best fits the mathematical model to the real system is continuously updated, improving dynamics prediction in the neighbourhood of the actual state.

Two are the main approaches to simultaneously estimate system state and parameters and they are referred to as dual and joint estimation. In the latter case parameters are treated as additional quasi-constant states and state estimation is carried out on

[1] This is more evident noticing that parameters may have more than one meaning: for instance, physical properties of the system and fitting coefficients of the model. The mathematical model is a representation of the physical system and cannot describe completely all its details, i. e. model inaccuracies are always present. Thus, we cannot expect system parameters with a physical meaning to assume values exactly equal to those properties estimated with other experiments.

an augmented state vector, stacking both state and parameters. Alternatively, dual estimation consists in running two state estimation processes in parallel: one working on state and the other on parameters, while updating parameters and state respectively at each step. In the following Sections 3.3 and 3.4, the dual estimation approach is exploited in order to deal only with (almost) linear systems. Instead, joint estimation would couple the dynamics of states and parameters, leading to non-linear systems.

Finally, two well-known estimators are the Kalman filter (KF) and its extensions, e. g. extended (EKF) and unscented (UKF) Kalman filters, and moving horizon estimation (MHE). The Kalman filter is the optimal state estimator for unconstrained, linear systems subject to normally distributed and independent state and measurement noises, while MHE is a computationally feasible online solution for state estimation based on an optimization strategy that accurately employs the nonlinear model and incorporates constraints into the optimization [66]. A discussion about numerical feasibility and methods for MHE is found in [67].

Outline Experimental setup currently available and procedures carried out during the project are described in §3.1, in order to discuss about which variables can be measured and then which are the input to estimation algorithms. Offline parameters identification developed in §3.2 follows an optimization-based approach, that resembles MHE. Model parameters are identified by applying this technique to real experimental data collected during this work. Another strategy, based on steady state approximation, is presented and discussed.

Then, we focus on state estimation algorithms based on Kalman filtering techniques. Values of parameters as found in the previous offline procedure may enter these algorithms as constant values or they may be iteratively updated to fit at best the real system behaviour; the latter case is a simultaneous state estimation and parameters identification. An algorithm based on KF for state estimation and parameters identification of the robot model presented in §2.1 is proposed in §3.3. Then, we analyse the problem of estimating droplet state and parameters for the model in §2.4. In particular, because of practical issues in measuring thermal gradients, a Kalman filter with unknown input (UI-KF) is presented in §3.4, that enables to find an approximate value for the input too. A procedure for liquid state estimation is proposed in §3.5, considering the FE model developed in §2.3.2. Moreover, we discuss about the high dimensionality of liquid state when FEM is adopted and no model reduction techniques are employed.

3.1 Experimental setup

This Section describes the apparatus and the procedures adopted during the experimental work carried out at the Hanczyc Lab (CIBIO, University of Trento) during this project (September 2016)[2].

The EvoBot, developed within the EVOBLISS project [43], is employed due to its layered structure, that allows to easily manage all the different parts of the system under analysis. Let us describe in more details the main components of the experimental apparatus.

Laser The device is taken from a *Laser Engraver Upgrade Pack for XY-Plotter Robot Kit* distributed by [68]. It is rigidly attached to the moving head of the EvoBot,

[2]The author would thank Silvia Holler for helping in carrying out experiments and recording videos and Amedeo Setti for droplet and laser tracking algorithm.

Table 3.1: Details of the experimental setup.

tilt angle	α_L	$30°$
laser power	p_L^{max}	420 mW
laser wavelength	λ_L	405 nm
liquid volume	V_L	9 mL
Petri dish inner radius	R_L	45 mm
laser emitter-liquid distance		40 mm
droplet volume	V	20 μL
cardboard hole diameter		85 mm
camera resolution		640×480 pixels
camera speed		8 fps

that can move on an horizontal plane. For what concerns these experiments, the head is actuated by hand (if needed), mainly along one axis only. The laser beam is directed downward and the device is tilted at α_L w.r.t. the vertical axis, in the vertical plane that comprises the main direction of motion. Focal distance, light power p_L, wavelength λ_L, input voltage are constant.

Liquid A volume of liquid medium is contained in a (circular) Petri dish, horizontal and flat. The container is on a cardboard positioned on the middle layer of the EvoBot; it is placed in the centre of a hole in the cardboard, such that the laser beam cannot engrave the support (hole diameter is slightly smaller than Petri dish diameter). Liquid volume V_L, Petri dish size and position are constant. The liquid is an aqueous solution of decanoate (concentration 5 mM, pH 11).

Droplet A single (lens shaped) droplet is formed on the free surface of the liquid layer by using a micro-pipette. Droplet liquid is 1-decanol, red coloured with Oil Red O. Before starting an experiment, typically the droplet is arranged in the centre of the Petri dish by using a plastic tip.

Camera An RGB camera is placed below the middle layer of the EvoBot. The hole in the cardboard permits to look at the droplet and the laser spot. The camera is a *Megapixel* $10\times$ *Digital Zoom* $f = 3.85$ mm and it is connected to a recording system via USB.

More detailed informations are reported in Tab. 3.1.

It is worth to notice that enough time passed between consecutive tests so that temperature field returned to be uniform and at room temperature. Moreover, enough time passed before starting each experiment to avoid waves and liquid motion and ensure the droplet to be still. Finally, effects of environmental air have been minimized by covering the EvoBot.

Let us focus now on the peculiarities of each experiment:

EXP1 robot head is hand-actuated along one axis only, trying to drive the droplet along that axis; liquid is decanoate solution;

EXP2 robot head is hand-actuated mainly but not only along one axis, trying to drive the droplet along that axis; liquid is decanoate solution;

EXP3 robot head is fixed; liquid is decanoate solution.

Experiments EXP1 and EXP2 show 1- and 2-dimensional laser paths and the response of the system, i. e. droplet paths, are expected to be complex due to the surrounding temperature field. In EXP3 a different scenario is set up, in which the laser beam does not change position, so that temperature field and consequent droplet path should be simpler.

Measurements Finally, let us consider which data may be available for parameters identification and state estimation. Robot dynamics is not involved in these experiments, so it cannot be identified. However, we expect the thermal dynamics of liquid medium to be excited by the input and the droplet to migrate accordingly. In particular, the model proposed in Chapter 2 should predict liquid and droplet behaviours.

The available experimental setup permits to measure laser spot positions during time, $\mathbf{y}^x(t)$, and droplet positions during time, $\mathbf{z}^x(t)$. This is made possible by a computer vision algorithm able to track objects from recorded video. Temperature field of liquid medium cannot be monitored over time, but it could be measured by a thermo-camera (not available yet). Initial and boundary conditions could be estimated by approximating to the room temperature.

3.2 Offline parameters identification

This Section discusses about how to estimate model parameters such that the mathematical model fits at best the physical system. This could be done by minimizing the error between model predictions and actual measurements. Then, the calibrated model is validated by comparing predictions and measurements of other tests.

Let us focus on the thermal model developed in §2.3.2 and droplet model proposed in §2.4. As explained in §3.1, robot dynamics cannot be identified by using available experimental data. In general, parameters may depend on time, but in the model calibration phase we consider them to be time-invariant.

$$\boldsymbol{M}_w(\mathbf{p})\dot{\mathbf{w}}(t) = -\boldsymbol{K}_w(\mathbf{p})\mathbf{w}(t) + \boldsymbol{B}_w(\mathbf{y}^x(t), \mathbf{p})\mathbf{u}^p(t) \tag{3.1}$$

$$\boldsymbol{M}_z(\mathbf{p})\dot{\mathbf{z}}(t) = \boldsymbol{A}_z(\mathbf{z}^v(t), \mathbf{p})\mathbf{z}(t) + \boldsymbol{B}_z(\mathbf{z}^x(t), \mathbf{p})\mathbf{w}(t) \tag{3.2}$$

Measurements give estimates of laser position $\hat{\mathbf{y}}^x$ and droplet position $\hat{\mathbf{z}}^x$ in a finite set of sampling times $\Gamma_m = \{0 \le t_{0,m} < t_{1,m} < \cdots < t_{M,m} \le T\}$. It is known when laser is switched on (full power) and off, but emitted power p_L is not *a priori* given, because it depends on the voltage. However, thanks to the linearity with coefficient β, we can assume the device to emit the maximum power and hence $\hat{\mathbf{u}}^p$ is available. For what concerns the temperature field, if initial temperature is uniform and equal to the constant uniform temperature at the boundary, namely

$$\begin{aligned} & \exists\, \bar{w} \in \mathbb{R} \\ \text{such that} \quad & \mathbf{w}(0) = \mathbf{1}\bar{w} \\ & \mathbf{w}_c(t) = \mathbf{1}\bar{w} \qquad \forall t \in [0, T] \end{aligned}$$

then it is not needed to have these values because they do not affect the substantial behaviour of the system. Thermal field is (constantly) biased but droplet response depends on temperature gradient, that is not biased. We assume this hypothesis to hold and consider w.l.o.g. a reference temperature $\bar{w} = 0\,^{\circ}\mathrm{C}$.

Let us formulate the parameters identification problem as a standard optimization problem; the optimal set of parameters should minimize the deviation between measurements and model predictions. Given vector of parameters $\mathbf{p}^*$ and initial droplet state $\mathbf{z}(0)$ (and initial and boundary conditions of temperature field), the dynamical model can be integrated forward in time to obtain the evolution[3] of temperature field $\mathbf{w}^*(t)$ and droplet state $\mathbf{z}^*(t)$ for $t \in [0, T]$. Then, based on simulated and actual measurements, a cost function can be defined to evaluate the optimality of parameters $\mathbf{p}^*$. Minimizing this cost function, we can identify parameters that fit the model to the physical system.

First, let us consider a continuous-time framework, hence we suppose to have control inputs $\hat{\mathbf{y}}^x(t)$ and $\hat{\mathbf{u}}^p(t)$ and measurements $\hat{\mathbf{z}}^x(t)$ for $t \in [0, T]$. The cost functional may be a quadratic cost on the different between laser-droplet distances obtained by model and measurement, d and $\hat{d}$ respectively. With discrete-time measurements $\hat{d}_i = \hat{d}(t_{i,m})$ for $i \in \{0, \dots, M\}$, this functional can be approximated with a quadrature rule, namely:

$$\mathcal{J}_{\mathrm{ID}} = \frac{1}{T} \int_0^T \|d(t) - \hat{d}(t)\|^2 \,\mathrm{d}t \approx \frac{1}{M+1} \sum_{i=0}^{M} \|d(t_{i,m}) - \hat{d}_i\|^2 \tag{3.3}$$

The optimal vector of parameters $\hat{\mathbf{p}}$ can be identified for $t \in [0, T]$ by minimizing the cost (3.3) while satisfying initial and boundary conditions and dynamic models (3.1)-(3.2). The minimizer $\hat{\mathbf{p}}$ is an estimate of model parameters that represent the actual physical system.

Notice that cost $\mathcal{J}_{\mathrm{ID}}$ in (3.3) presumes the presence of noise in the measurement only. Actually, process noise is unavoidable and it may have a significant effect on the evolution of the system. Introducing a continuous-time disturbance $\boldsymbol{\nu}$, only in droplet dynamics for simplicity, the aforementioned optimization problem turns into an OCP, defined as:

$$\begin{aligned}
\text{minimize} \quad & \mathcal{J}_{\mathrm{ID}} + \frac{1}{T} \int_0^T \omega_\nu \|\boldsymbol{\nu}(t)\|^2 \,\mathrm{d}t \\
\text{subject to} \quad & \boldsymbol{M}_w(\mathbf{p})\, \dot{\mathbf{w}}(t) = -\boldsymbol{K}_w(\mathbf{p}) \mathbf{w}(t) + \boldsymbol{B}_w(\hat{\mathbf{y}}^x(t), \mathbf{p})\, \hat{\mathbf{u}}^p(t) \\
& \boldsymbol{M}_z(\mathbf{p})\, \dot{\mathbf{z}}(t) = \boldsymbol{A}_z(\mathbf{z}^v(t), \mathbf{p})\, \mathbf{z}(t) + \boldsymbol{B}_z(\mathbf{z}^x(t), \mathbf{p})\, \mathbf{w}(t) + \boldsymbol{\nu}(t) \\
& \mathbf{w}(0) = \mathbf{1}\bar{w} \\
& \mathbf{w}_{\mathrm{c}}(t) = \mathbf{1}\bar{w} \\
& \mathbf{z}(0) = \hat{\mathbf{z}}(0) \\
\text{for} \quad & t \in (0, T)
\end{aligned} \tag{3.4}$$

where the hat denotes measured quantities. Problem (3.4) resembles a standard OCP with control cost, where $\boldsymbol{\nu} = \boldsymbol{\nu}(t)$ is the control (of dimension n_ν). This is quite different w.r.t. the problem cited above, that is a (static) optimization problem with few optimization variables—n_p, the number of parameters in vector $\mathbf{p}$. Instead, after

[3] Sensitivity of the evolution to initial conditions and model parameters is emphasized by the unstable dynamics of the system, that may be considered chaotic.

time discretization over a grid with n_t nodes, Problem (3.4) involves $(n_p + n_t n_\nu)$ optimization variables, leading to a high (finite) dimensional problem.

Let us focus on which are the unknown parameters. Mass matrix $\boldsymbol{M}_w$ depends (linearly) on specific mass ϱ, specific heat c_p and thickness l_z of the liquid layer; stiffness matrix $\boldsymbol{K}_w$ depends (linearly) on its thermal conductivity k and thickness l_z; nodal loads vector depends on coefficient β (linearly)[4] and characteristic radius r_L of the laser beam intensity profile. Matrix $\boldsymbol{M}_z$ is given by droplet mass m_D; state matrix $\boldsymbol{A}_z$ depends on coefficient c_D, input matrix $\boldsymbol{B}_z$ comprises coefficient b_D and linear operator $B(\mathbf{z})$. Thermal properties of the liquid medium may be estimated considering its chemical composition and temperature. Liquid layer thickness l_z is related to the poured volume of liquid V_L and container shape and size. Droplet mass m_D is given by its specific mass ϱ' and volume V. Remaining parameters $\{\beta, r_L, c_D, b_D\}$ are unknowns.

Let us discuss the consequences of having no measurements of the temperature field. First, it is important to notice that (3.1) is a linear ODE subject to homogeneous initial conditions and constant homogeneous boundary conditions in Problem (3.4), thus temperature gradient is linear w.r.t. coefficient β[5]. Temperature gradient affects droplet dynamics (3.2) through the input matrix $\boldsymbol{B}_z$ that depends linearly on parameter b_D. Thus, from the droplet point-of-view, parameters β and b_D have the same effect; with no informations about the temperature field, these parameters are undistinguishable. Finally, parameters $\{\beta b_D, r_L, c_D\}$ could be estimated.

Droplet spatial extension is not considered in the model proposed in §2.4. This modelling inaccuracy likely leads to prediction errors and difficulties in parameters identification because of this missing part in the model, especially when the laser beam is close to the droplet. Here we propose to add an auxiliary parameter δ to take into account this inaccuracy. In particular, the point-mass droplet is translated w.r.t. the physical droplet along the line connecting laser beam and droplet itself. Effective distance between laser beam and point-mass droplet is likely a function of distance itself; we restrict to the much simpler case of constant δ. In particular, droplet initial position changes, namely:

$$\mathbf{z}_0^x \leftarrow \mathbf{z}_0^x - \delta \frac{\mathbf{z}_0^x - \mathbf{y}_0^x}{\|\mathbf{z}_0^x - \mathbf{y}_0^x\|} \tag{3.5}$$

Then, evolution is simulated and laser-droplet distance d is computed and added to the bias δ, i. e. $d \leftarrow d + \delta$, before evaluating the aforementioned cost function.

Finally, initial droplet velocity $\mathbf{z}_0^v$ is added to optimization variables because its estimation from measurements may be rough (as initial position is) but its value may strongly affect the evolution of the system. A guess for $\mathbf{z}_0^v$ can be computed by using the finite difference approximation. Hence, unknown model parameters are $\{\beta b_D, c_D, r_L, \delta, \mathbf{z}_0^v\}$.

Numerical solution Problem (3.4) cannot be directly solved by using any numerical methods. Following the direct method (with full discretization), a time grid, a state discretization scheme and a control parametrization scheme are introduced to obtain a finite dimensional (perhaps nonlinear) optimization problem [6, 27], also called NLP.

The time grid for state discretization, $\Gamma = \{0 \le t_0 < t_1 < \cdots < t_N \le T\}$, is independent on measurement time grid Γ_m, because it is used only to predict

[4] Nodal loads depend on incident power too, that is not known nor estimated. We can consider $p_L = p_L^{max}$ and coefficient β accounts also for power losses.

[5] Outline of the proof: subdivide free and constrained nodes; express the system in state-space form with temperature gradient as output; Laplace transform and solve for the temperature gradient.

the evolution of the system. In order to reproduce accurately the continuous-time dynamics, the maximum time step in Γ should be small enough, depending on the discretization scheme. Grid Γ is likely more refined than Γ_m, namely $\forall i \in \{0, \dots, M-1\}$ $\exists (j,k) \in \{0, \dots, N-1\} \times \{1, 2, \dots\}$ such that:

$$t_{i,m} \leq t_j < \cdots < t_{j+k} \leq t_{i+1,m} \tag{3.6}$$

Moreover, we build grid Γ such that measurement times are nodes, so that interpolation is not needed to compute predictions at those times.

$$t_{i,m} \in \Gamma \quad \forall i \in \{0, 1, \dots, M\} \tag{3.7}$$

We adopt a central difference scheme for thermal dynamics, as presented in §2.3.2, and an approximated central difference scheme for droplet dynamics, derived in §3.4. Controls are parametrized with piece-wise linear functions (trapezoidal quadrature rule is used).

A this step a nonlinear optimization problem (NLP) with quadratic cost function and (possibly) bounds on optimization variables[6] has to be solved. This NLP may be large if Problem (3.4) is considered as it is, and its dimensionality increases with the number of nodes of Γ. Instead, Problem (3.4) with imposed $\boldsymbol{\nu}(t) = 0$, $t \in [0, T]$, has n_p optimization variables for every choice of the time grid Γ; let us refer to this as the *reduced* Problem (3.4). Results reported in §3.2.1 are obtained by facing this reduced NLP and by using the `lsqnonlin` routine, that is a non-linear least-squares solver provided by Matlab.

3.2.1 Data and results

In this Section the dynamical model (3.1)-(3.2) is calibrated starting from experimental data collected with the setup described in §3.1. In particular, experiment EXP3 is adopted as training set of data, i. e. for parameters identification, and EXP1-EXP2 as validation sets, i. e. for model validation.

Let us estimate physical properties and coefficients to be used as fixed parameters in the mathematical model. These data are reported in Tab. 3.2. First of all, for what concerns thermal properties we assume that pure water and decanoate solution have the same properties. This hypothesis is reasonable because the solution is dilute; concentration 5 mM approximately corresponds to 9×10^{-5} mol/mol of decanoate in water. Surface tension is likely different for the two solutions, however that of water is considered. In particular, surface tension σ of water depends on temperature T as $\sigma(T) = B\tau^{\mu}(1 + b\tau)$, where $\tau = 1 - T/T_c$, $T_c = 647\,\mathrm{K}$, $B = 236\,\mathrm{mN/m}$, $b = -0.625$ and $\mu = 1.26$ [69]. Then, surface tension coefficient σ_T can be approximated as $\frac{\partial \sigma}{\partial T}(T)$ evaluated at room temperature $T \approx 20\,^{\circ}\mathrm{C}$, yielding $\sigma_T = -0.15\,\mathrm{mN/m\,K}$.

Then, liquid layer thickness is computed by considering the volume of a cylinder inside the Petri dish. Droplet radius has been roughly measured, $R \approx 3\,\mathrm{mm}$, and hence shape factor can be estimated to be $\varepsilon = 0.2$ considering droplet volume, so that droplet thickness is approximately $\varepsilon R = 0.6\,\mathrm{mm} \approx l_z/2$, that is reasonable.

Guess An initial guess, i. e. an estimate of parameters, is typically needed to initialize the optimization routine. In general, more accurate guesses give faster and better

[6] Parameters in $\mathbf{p}$ may be bounded between *a priori* estimated values. Process noise at each time node $\boldsymbol{\nu}_i$, $i \in \{0, \dots, N\}$, is independent (white noise is assumed) and should be realistically bounded, depending on the time step size.

Table 3.2: Physical properties and parameters of liquid layer and droplet.

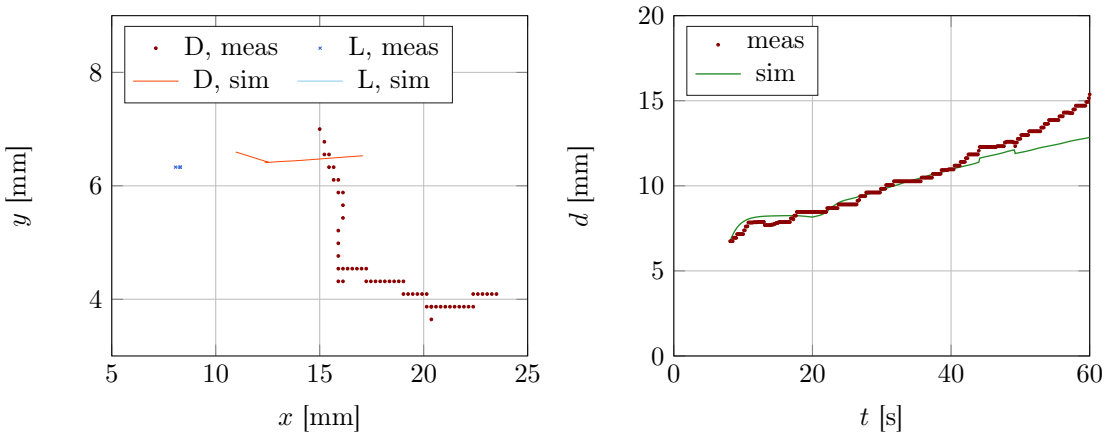
ϱ'	830	$\mathrm{kg/m^3}$
ϱ	997	$\mathrm{kg/m^3}$
c_p	4186	$\mathrm{J/kg\,K}$
k	0.6	$\mathrm{W/m\,K}$
σ_T	-0.15	$\mathrm{mN/m\,K}$
l_z	1.4	mm
ε	0.2	-
R	2.9	mm
A	5.2	$\mathrm{mm^2}$

Table 3.3: Parameters identification of EXP3: initial guess.

Parameters		
c_x	0.005	-
c_1	201	-
β	70.7	$\mathrm{m^{-1}}$
Guess		
$b_D\beta$	-1.77×10^{-5}	N/K
c_D	1.30×10^{-5}	kg/m
r_L	1.5	mm
δ	3.46	mm
$\mathbf{z}_0^v$	$[-0.7, 1]$	mm/s

solutions. Let us find a reasonable approximation of parameters $\{\beta b_D, c_D, r_L, \delta, \mathbf{z}_0^v\}$. Absorption coefficient β, as defined in §2.2, can be estimated considering a percentage of absorption p_β and the layer thickness l_z, namely $\beta = p_\beta / l_z$. In the following it is $p_\beta = 0.1$ due to the transparency and small thickness of the liquid layer. Input gain b_D and state gain c_D can be estimated by recalling the model of thermocapillary motion proposed in §2.4, namely $b_D = c_1 R^2 \sigma_T$ and $c_D = \varrho A c_x / 2$. Initial droplet velocity $\mathbf{z}_0^v$ is approximated by using finite differences. Laser beam radius r_L and bias δ have been tuned; notice that it is $\delta = 1.2R$, i. e. the (virtual) point-mass droplet is almost on the (real) droplet edge. Initial guess for parameters identification of EXP3 is reported in Tab. 3.3.

Solution Results of the optimization process are discussed in this paragraph. Table 3.4 reports the values of identified parameters. It is worth to recall that these values may correspond to a local minimum for the reduced Problem 3.4. Predicted and measured paths of droplet, i. e. $\mathbf{z}$ and $\hat{\mathbf{z}}$, are shown in Fig. 3.1a with laser positions. Bias due to δ is clearly seen, because initially simulated droplet is not placed in the measured position. Then, it is also evident that paths do not match. However, looking at Fig. 3.1b, the model prediction fits measurements quite well, when expressed in terms of laser-droplet distance. This metrics is less sensitive to noise and initial conditions



(a) *Paths.* **(b)** *Distance.*

Figure 3.1: Parameters identification of EXP3: measurements and predictions.

w.r.t. the deviation in the absolute position of the droplet, for instance. Then, it has to be highlighted the increasing of the error $\|d - \hat{d}\|$ after a long heating time interval — starting at about $t \approx 45\,\mathrm{s}$ in Fig. 3.1b. This deviation may be due to a modelling inaccuracy: droplet could be driven by convective flows arisen because of thermocapillary forces. This fact may limit the region of validity and applicability of the model proposed in this work.

Table 3.4: Parameters identification of EXP3: optimization results.

$b_D\beta$	-5.31×10^{-6}	N/K
c_D	1.27×10^{-5}	kg/m
r_L	1.02	mm
δ	4.06	mm
$\mathbf{z}_0^v$	$[-0.13, 1.16]^T$	mm/s

3.2.2 Model and parameters validation

Identified parameters, in particular $\{b_D\beta, c_D, r_L\}$ reported in Tab. 3.4, should be able to predict the system evolution for other tests too. Different values for δ and $\mathbf{z}_0^v$ are allowed: the former because bias may be a function of laser-droplet distance, as explained above, and the latter because initial conditions typically do not coincide. Thus, solving reduced Problem 3.4 with parameters $\{b_D\beta, c_D, r_L\}$ fixed with identified values, i. e. finding only optimal bias and initial velocity, the model prediction should match measurements. This procedure could be considered a validation of the model.

Measurements obtained by EXP1 and EXP2 are employed to this purpose. These tests share the same setup of EXP3 and only the control input is different, i. e. the sequence of laser positions, thus model parameters should coincide and the proposed procedure is meaningful. Results are shown in graphical form in Fig. 3.2 and Fig. 3.3;

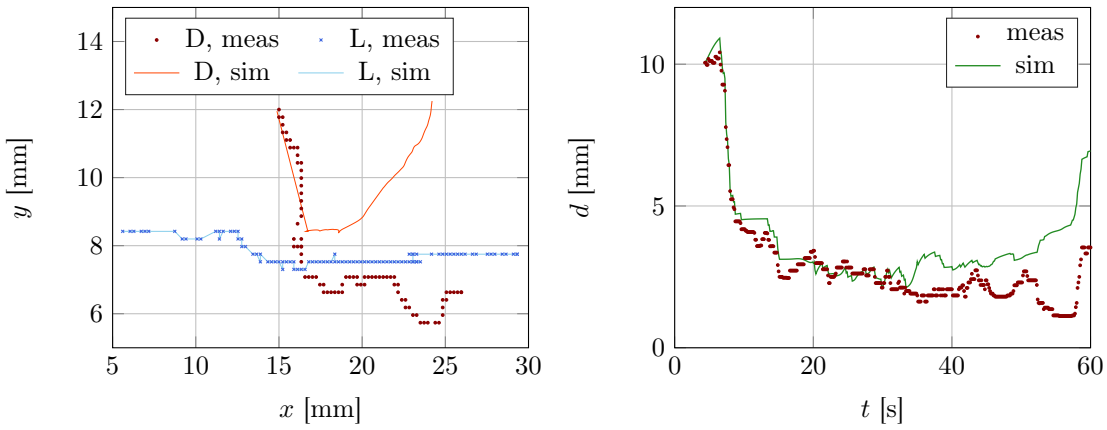
corresponding bias δ and initial droplet velocity $\mathbf{z}_0^v$ are reported in tabular form in Tab. 3.5.

Predictions and measurements match in a time interval and then deviation increases due to system instability and sensitivity to noise[7]. However, dynamics of the model fits the real evolution, considering short time intervals — about $\Delta t \approx 30\,\text{s}$.

Finally, we mention that the `patternsearch` function provided by Matlab has been adopted to solve the optimization problem introduced in this Section. Using a direct search algorithm, it is possible to explore a search region without getting stuck in local minima. Values of initial droplet velocity $\mathbf{z}_0^v$ change for different tests, as expected. Notice that also bias δ in Tab. 3.4 and Tab. 3.5 differ; this fact was expected and announced in §3.2. So, it has to be underlined that the proposed point-mass model is not able to represent the effect of droplet extension.

Table 3.5: Model validation: optimization results.

	EXP1	EXP2	
δ	0.13	1.35	mm
$\mathbf{z}_0^v$	$[-2.70, 1.42]^T$	$[-0.52, 1.98]^T$	mm/s



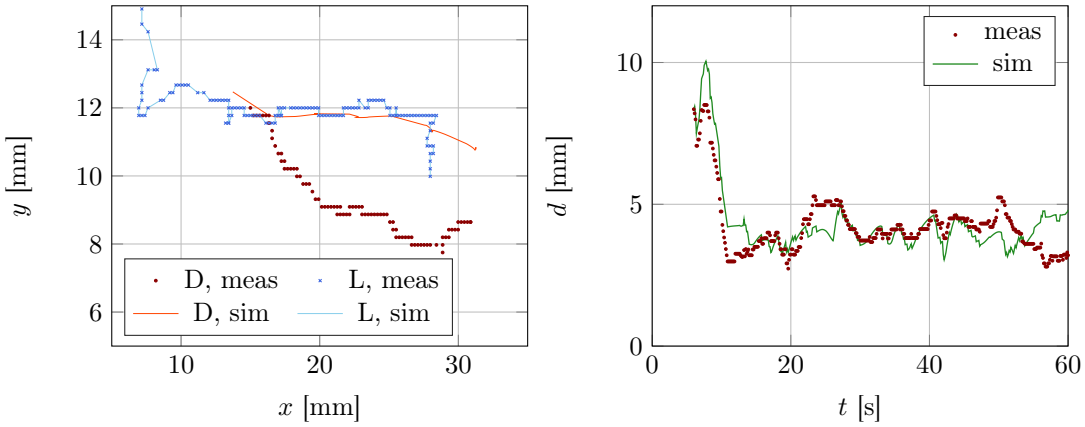
(a) *Paths.* **(b)** *Distance.*

Figure 3.2: Model validation with EXP1.

3.2.3 Steady state identification

Different experimental procedures could be adopted to obtain data sets suitable for parameters identification. Here we suggest a method based on the analytical solution found in §2.3.1 for a continuous moving heat source. The idea is to bring the real system into a steady state condition and identify the parameters that characterize the steady state component of the dynamical model.

[7] Analysis of error distribution is not a useful tool to evaluate residuals of the fitting in this case. This is due to the unstable dynamics of the model: noise strongly affects the evolution and then the future error, so errors are correlated.



(a) *Paths.* **(b)** *Distance.*

Figure 3.3: Model validation with EXP2.

Let us consider w.l.o.g. the 1-dimensional case. Therefore, after a transient phase, the system could be described by the relative position ξ_{ss} between droplet and laser beam and their absolute speed v_{ss}—that has to be the same, because ξ_{ss} is constant in steady state conditions.

Let us recall thermal model (2.49) and droplet dynamics (2.88) and consider both temperature field and motion along the x axis of a fixed reference frame. The robot moves the laser at speed $v_L > 0$, such that $x_L(t) = v_L t$. The relative position of a point w.r.t. the laser beam is $\xi(t,x) = x - x_L(t)$. This leads to the system:

$$w(t,x) = \bar{w} + \frac{\beta p_L}{2\pi k}\mathcal{K}_0\left(\frac{v_L}{2\alpha}|\xi(t,x)|\right)e^{-v_L\xi(t,x)/(2\alpha)} \tag{3.8}$$

$$\dot{x}_D(t) = v_D(t) \tag{3.9}$$

$$m_D\dot{v}_D(t) = b_D\frac{\partial w}{\partial x}(t,x_D(t)) - c_D|v_D(t)|v_D(t) \tag{3.10}$$

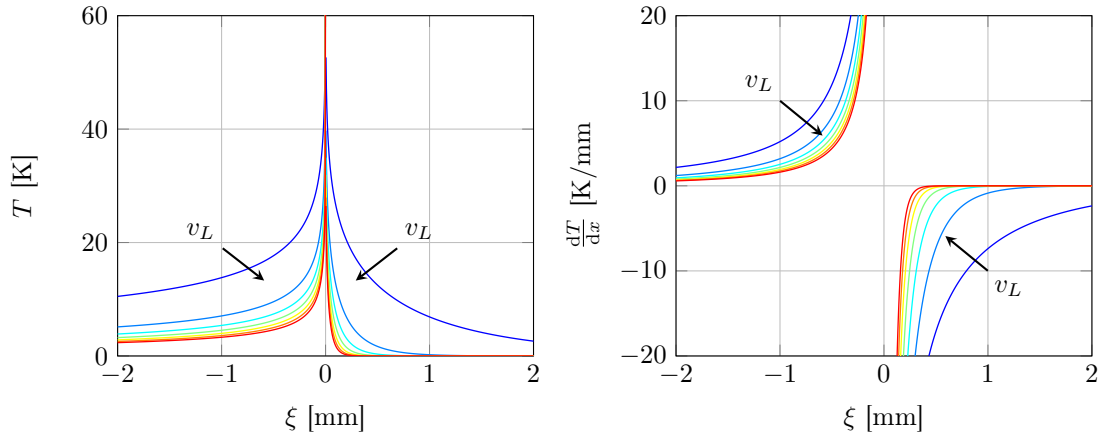
Temperature gradient can be explicitly derived from (3.8); denoting $\sigma = v_L\xi/(2\alpha)$ for simplicity, it is:

$$\frac{\partial w}{\partial x}(t,x) = -\frac{v_L}{2\alpha}\frac{\beta p_L}{2\pi k}e^{-\sigma(t,x)}\left(\mathcal{K}_1\left(|\sigma(t,x)|\right)\operatorname{sign}(\sigma(t,x)) + \mathcal{K}_0\left(|\sigma(t,x)|\right)\right) \tag{3.11}$$

Focusing on steady state conditions, the constraints $v_L = v_{ss}$ and $v_D(t)|_{ss} = v_{ss}$ have to be enforced. Eq. (3.10) gives that temperature gradient in droplet position has to be constant (as expected in steady state conditions):

$$\begin{gathered} 0 = b_D\frac{\partial w}{\partial x}(t,x_D(t))\Big|_{ss} - c_D|v_{ss}|v_{ss} \\ \Downarrow \\ \frac{\partial w}{\partial x}(t,x_D(t))\Big|_{ss} = \frac{c_D}{b_D}|v_{ss}|v_{ss} = w'_{ss} \end{gathered} \tag{3.12}$$

Actually, this is true also because the distance between droplet and laser does not change in steady state conditions. Substituting (3.9) and integrating (integration



(a) *Temperature.* **(b)** *Temperature gradient.*

Figure 3.4: Temperature profile along the direction of motion of a continuous moving point-wise heating source.

constant ξ_0) we find that:

$$\xi(t, x_D(t))\Big|_{ss} = x_D(t)\Big|_{ss} - v_{ss}t = v_{ss}t + \xi_0 - v_{ss}t = \xi_0 = \xi_{ss} \tag{3.13}$$

and hence $\sigma_{ss} = v_{ss}\xi_{ss}/(2\alpha)$. Notice that also temperature in droplet position is constant; from (3.8) it is:

$$w(t, x_D(t))\Big|_{ss} = \bar{w} + \frac{\beta p_L}{2\pi k}\mathcal{K}_0\left(|\sigma_{ss}|\right)e^{-\sigma_{ss}} = w_{ss} \tag{3.14}$$

Stationary temperature gradient w'_{ss} in (3.12) has to satisfy the thermal model (3.11) too, namely the condition:

$$w'_{ss} = \frac{\partial w}{\partial x}(t, x_D(t))\Big|_{ss} = -\frac{v_{ss}}{2\alpha}\frac{\beta p_L}{2\pi k}e^{-\sigma_{ss}}\left(\mathcal{K}_1\left(|\sigma_{ss}|\right)\operatorname{sign}(\sigma_{ss}) + \mathcal{K}_0\left(|\sigma_{ss}|\right)\right) \tag{3.15}$$

Then, from Equations (3.12) and (3.15) we can get rid of the steady state temperature gradient w'_{ss}, and a relationship between v_{ss}, σ_{ss} and unknown parameters is found.

$$\frac{c_D}{b_D\beta} = -\frac{1}{|v_{ss}|}\frac{p_L}{4\pi k\alpha}e^{-\sigma_{ss}}\left[\mathcal{K}_1\left(|\sigma_{ss}|\right)\operatorname{sign}(\sigma_{ss}) + \mathcal{K}_0\left(|\sigma_{ss}|\right)\right] \tag{3.16}$$

From experimental data and *a priori* knowledge steady state values v_{ss} and ξ_{ss}, thermal properties α and k of the liquid and laser power p_L can be estimated. Hence, the ratio $c_D/(b_D\beta)$ can be computed by adopting the definition of σ_{ss} and Eq. (3.16). The estimate of this ratio should be compared with results of §3.2.1.

Analysis Let us discuss about the applicability of this approach. First of all, if laser beam intensity profile is Gaussian and its radius is not small, then the analytical solution derived in §2.3.1 may give inaccurate results, because it assumes a point-wise heating source. Point-mass model of droplet is an useful simplification, but this may

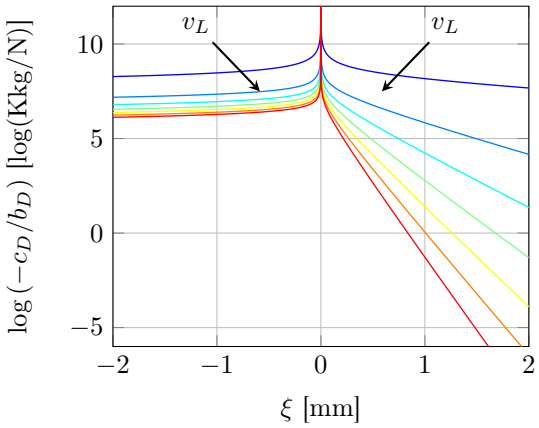


Figure 3.5: Ratio c_D/b_D and steady state condition: sensitivity analysis for point-wise heating source.

require to compensate steady state relative position ξ_{ss} with bias δ, that actually could depend on ξ_{ss} itself.

Finally, a significant issue could be the robustness of this method. Fig. 3.4 shows profiles of temperature and temperature gradient in space as a function of laser velocity. These functions are so sharp due to the point-wise heating source; in the case of Gaussian intensity profile we expect these maps to be more bounded — no asymptotes and lower gradients. However, this method could be much sensitive to noise, leading to large errors in the ratio $c_D/(b_D\beta)$. Also notice that Gaussian white noise added to data would return a non-Gaussian error distribution in estimated ratio, because function at the r.h.s. of Eq. (3.16) is non-linear. Sensitivity analysis depicted in Fig. 3.5 shows an exponential dependence of ratio c_D/b_D to steady state velocity and distance, for a given absorption coefficient β.

3.3 Robot dual estimation using Kalman Filter

This Section analyses a procedure based on Kalman filtering for the dual estimation of state and parameters of the robotic platform. Let us consider the dynamical model of the robotic platform proposed in §2.1 and add dynamics of parameters and measurement equation, neglecting disturbances; for $t \in [0, T]$ it is:

$$\begin{aligned} \boldsymbol{M}_y(\mathbf{p}(t))\dot{\mathbf{y}} &= \boldsymbol{A}_y(\mathbf{p}(t))\mathbf{y}(t) + \boldsymbol{B}_y(\mathbf{p}(t))\mathbf{u}(t) \\ \dot{\mathbf{p}}(t) &= \mathbf{0} \\ \mathbf{m}(t) &= \boldsymbol{C}_y\mathbf{y}(t) \end{aligned} \tag{3.17}$$

where $\mathbf{y}$ is the robot state vector, $\mathbf{p}$ the robot parameters, $\mathbf{u}$ the control actions, $\mathbf{m}$ the measurement.

Forward difference Let us build a time grid $\Gamma = \{0 = t_0 < t_1 < \cdots < t_N = T\}$ and apply the forward difference scheme, also known as explicit Euler method, for time discretization, namely for $k = 1, 2, \ldots, N$:

$$\frac{\boldsymbol{M}_y(\mathbf{p}_k)\mathbf{y}_k - \boldsymbol{M}_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1}}{\Delta t_k} = \boldsymbol{A}_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1} + \boldsymbol{B}_y(\mathbf{p}_{k-1})\mathbf{u}_{k-1} \tag{3.18}$$

where the discrete-time state approximation is denoted $\mathbf{y}(t_k) \approx \mathbf{y}_k$ and time step size by $\Delta t_k = t_k - t_{k-1}$. Constant dynamics of parameters, $\mathbf{p}_k \approx \mathbf{p}_{k-1}$, leads to the following explicit scheme:

$$\begin{aligned} \boldsymbol{M}_y(\mathbf{p}_{k-1})\frac{\mathbf{y}_k - \mathbf{y}_{k-1}}{\Delta t_k} &= \boldsymbol{A}_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1} + \boldsymbol{B}_y(\mathbf{p}_{k-1})\mathbf{u}_{k-1} \\ \Downarrow \\ \boldsymbol{M}_y(\mathbf{p}_{k-1})\mathbf{y}_k &= \boldsymbol{M}_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1} + \Delta t_k \left[\boldsymbol{A}_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1} + \boldsymbol{B}_y(\mathbf{p}_{k-1})\mathbf{u}_{k-1}\right] \end{aligned} \tag{3.19}$$

Collecting matrices into $\boldsymbol{A}_y^*$ and $\mathbf{b}_y^*$ and adding process Gaussian independent noise $\boldsymbol{\nu}_k^y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k^y)$ for $k \in \{0, \dots, N\}$, we get the linear time-varying discrete-time dynamics of robot state $\mathbf{y}$. Update equation of parameters and measurement with noises $\boldsymbol{\nu}_k^p \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k^p)$ and $\boldsymbol{\nu}_k^m \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_k)$ are easily obtained. Control input $\mathbf{u}_k$ is considered to be known without uncertainty for $k \in \{0, \dots, N\}$. The stochastic discrete-time model of the system is:

$$\begin{aligned} \mathbf{y}_k &= \boldsymbol{A}_y^*(\mathbf{p}_{k-1})\mathbf{y}_{k-1} + \mathbf{b}_y^*(\mathbf{p}_{k-1}, \mathbf{u}_{k-1}) + \boldsymbol{\nu}_{k-1}^y \\ \mathbf{p}_k &= \mathbf{p}_{k-1} + \boldsymbol{\nu}_{k-1}^p \\ \mathbf{m}_k &= \boldsymbol{C}_y\mathbf{y}_k + \boldsymbol{\nu}_k^m \end{aligned} \tag{3.20}$$

In order to run a parameter estimation process we need to express the measurement equation as a function of system parameters. Recalling matrices of the robotic platform model in §2.1, we find:

$$\boldsymbol{M}_y^{-1}\boldsymbol{A}_y = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ & -c_L/m_L\boldsymbol{I} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{M}_y^{-1}\boldsymbol{B}_y = \begin{bmatrix} \mathbf{0} \\ b_L/m_L\boldsymbol{I} \end{bmatrix} \tag{3.21}$$

Let us now explicit the updated state $\mathbf{y}_k$ as function of parameter $\mathbf{p} = [m_L, c_L, b_L]^T$. Partitioning state $\mathbf{y} = [\mathbf{y}^x, \mathbf{y}^v]^T$ and control $\mathbf{u} = [\mathbf{u}^p, \mathbf{u}^i]^T$, it is:

$$\begin{aligned} \mathbf{y}_k &= \mathbf{y}_{k-1} + \Delta t_k \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ & -c_L/m_L\boldsymbol{I} \end{bmatrix} \mathbf{y}_{k-1} + \Delta t_k \begin{bmatrix} \mathbf{0} \\ b_L/m_L\boldsymbol{I} \end{bmatrix} \mathbf{u}_{k-1} \\ &= \begin{bmatrix} \mathbf{y}_{k-1}^x + \Delta t_k \mathbf{y}_{k-1}^v \\ \mathbf{y}_{k-1}^v \end{bmatrix} + \Delta t_k \begin{bmatrix} \mathbf{0} & \\ -\mathbf{y}_{k-1}^v & \mathbf{u}_{k-1}^i \end{bmatrix} \begin{bmatrix} c_L/m_L \\ b_L/m_L \end{bmatrix} \end{aligned} \tag{3.22}$$

Then, defining parameter $\tilde{\mathbf{p}} = [c_L/m_L, b_L/m_L]^T$ we get the linear time-varying measurement equation from the robot parameter point-of-view:

$$\begin{aligned} \mathbf{m}_k &= \boldsymbol{C}_y\mathbf{y}_k + \boldsymbol{\nu}_k^m \\ &= \boldsymbol{C}_y \Delta t_k \begin{bmatrix} \mathbf{0} & \\ -\mathbf{y}_{k-1}^v & \mathbf{u}_{k-1}^i \end{bmatrix} \tilde{\mathbf{p}}_{k-1} + \boldsymbol{C}_y \begin{bmatrix} \mathbf{y}_{k-1}^x + \Delta t_k \mathbf{y}_{k-1}^v \\ \mathbf{y}_{k-1}^v \end{bmatrix} + \boldsymbol{C}_y\boldsymbol{\nu}_{k-1}^y + \boldsymbol{\nu}_k^m \\ &= \boldsymbol{C}_y^*(\mathbf{y}_{k-1}, \mathbf{u}_{k-1})\tilde{\mathbf{p}}_k + \mathbf{d}_y^*(\mathbf{y}_{k-1}) + \boldsymbol{C}_y\boldsymbol{\nu}_{k-1}^y + \boldsymbol{\nu}_k^m \end{aligned} \tag{3.23}$$

It is worth noticing that values of parameters cannot be determined, but only their ratios. But both robot state and parameters evolution and measurement are described by linear time-varying dynamics if the continuous-time model is discretized with a forward difference scheme.

Central difference Let us consider now the case of central difference scheme, that has a second order accuracy. Applying the same procedure adopted in §2.3.2, based on

trapezoidal quadrature rule and finite difference approximation, it is:

$$\frac{\boldsymbol{M}_y(\mathbf{p}_k)\mathbf{y}_k - \boldsymbol{M}_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1}}{\Delta t_k} = \frac{\boldsymbol{A}_y(\mathbf{p}_k)\mathbf{y}_k + \boldsymbol{A}_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1}}{2} + \\ + \frac{\boldsymbol{B}_y(\mathbf{p}_k)\mathbf{u}_k + \boldsymbol{B}_y(\mathbf{p}_{k-1})\mathbf{u}_{k-1}}{2} \tag{3.24}$$

Recalling the constant parameters dynamics, the update law reduces to:

$$\boldsymbol{M}_y(\mathbf{p}_{k-1})\frac{\mathbf{y}_k - \mathbf{y}_{k-1}}{\Delta t_k} = \boldsymbol{A}_y(\mathbf{p}_{k-1})\frac{\mathbf{y}_k + \mathbf{y}_{k-1}}{2} + \boldsymbol{B}_y(\mathbf{p}_{k-1})\frac{\mathbf{u}_k + \mathbf{u}_{k-1}}{2}$$
$$\Downarrow$$
$$\left(\boldsymbol{M}_y(\mathbf{p}_{k-1}) - \frac{\Delta t_k}{2}\boldsymbol{A}_y(\mathbf{p}_{k-1})\right)\mathbf{y}_k = \left(\boldsymbol{M}_y(\mathbf{p}_{k-1}) + \frac{\Delta t_k}{2}\boldsymbol{A}_y(\mathbf{p}_{k-1})\right)\mathbf{y}_{k-1} + \\ + \Delta t_k \boldsymbol{B}_y(\mathbf{p}_{k-1})\frac{\mathbf{u}_k + \mathbf{u}_{k-1}}{2} \tag{3.25}$$

that is again a linear time-varying dynamics for robot state $\mathbf{y}$; it can be written in a more compact form as:

$$\mathbf{y}_k = \boldsymbol{A}^*_y(\mathbf{p}_{k-1})\mathbf{y}_{k-1} + \mathbf{b}^*_y(\mathbf{p}_{k-1}, \mathbf{u}_{k,k-1}) + \boldsymbol{\nu}^y_{k-1} \tag{3.26}$$

As before, let us consider the measurement equation for parameter identification, first expressing the state update equation as a function of parameter $\mathbf{p}$. We have matrices

$$\boldsymbol{M}_y - \frac{\Delta t_k}{2}\boldsymbol{A}_y = \begin{bmatrix} \boldsymbol{I} & -\frac{\Delta t_k}{2}\boldsymbol{I} \\ & (m_L + \frac{\Delta t_k}{2}c_L)\boldsymbol{I} \end{bmatrix} \\ \boldsymbol{M}_y + \frac{\Delta t_k}{2}\boldsymbol{A}_y = \begin{bmatrix} \boldsymbol{I} & \frac{\Delta t_k}{2}\boldsymbol{I} \\ & (m_L - \frac{\Delta t_k}{2}c_L)\boldsymbol{I} \end{bmatrix} \tag{3.27}$$

Being mass m_L, friction coefficient c_L and time step Δt_k positive values, we can always invert and find the updated state $\mathbf{y}_k$ as a function of parameters. Using a central difference scheme for time discretization, it turns out that this relation is non-linear for $\mathbf{p}$ and $\tilde{\mathbf{p}}$.

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}^x_{k-1} \\ \mathbf{y}^v_{k-1} \end{bmatrix} + \frac{2m_L\Delta t_k}{2m_L + c_L\Delta t_k}\begin{bmatrix} \mathbf{y}^v_{k-1} \\ -\frac{c_L}{m_L}\mathbf{y}^v_{k-1} \end{bmatrix} + \frac{2b_L\Delta t_k}{2m_L + c_L\Delta t_k}\begin{bmatrix} \mathbf{0} & \frac{\Delta t_k}{2}\boldsymbol{I} \\ & \boldsymbol{I} \end{bmatrix}\frac{\mathbf{u}_k + \mathbf{u}_{k-1}}{2} \tag{3.28}$$

Procedure Robot state discrete-time update rule is linear time-varying adopting both forward and central difference scheme for time discretization. State measurement and parameters update rule are linear time-invariant. Measurement as function of parameters is linear time-varying in the case of forward difference but becomes non-linear with central difference scheme.

Based on these considerations, we propose a procedure for robot state and parameter dual estimation, based on linear Kalman filtering. Pseudo-code of (Extended) Kalman filter (KF or EKF) is reported in Algorithm 1. Robot state discrete-time dynamics (3.26) and measurement from parameters (3.23) are used in Algorithm 2. Notice that dual estimation does not take into account the covariance between state and parameters.

Algorithm 1: (Extended) Kalman Filter

Data: $\hat{\mathbf{x}}_{k-1}$, $\boldsymbol{P}_{k-1}$, $\mathbf{b}_{k-1}$, $\mathbf{y}_k$

// prediction step

if *model is linear* **then**

  $\hat{\mathbf{x}}_k^- = \boldsymbol{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{b}_{k-1}$; // state equation

  $\hat{\mathbf{y}}_k = \boldsymbol{C}_k\hat{\mathbf{x}}_k + \mathbf{d}_k$; // measurement equation

else

  $\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{b}_{k-1})$;

  $\hat{\mathbf{y}}_k = h(\hat{\mathbf{x}}_k, \mathbf{d}_k)$;

  $\boldsymbol{A}_{k-1} = \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{b})\Big|_{\hat{\mathbf{x}}_{k-1},\, \mathbf{b}_{k-1}}$; // linearisation

$\boldsymbol{P}_k^- = \boldsymbol{A}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{A}_{k-1}^T + \boldsymbol{Q}_{k-1}$;

// Kalman gain

$\boldsymbol{\mathcal{K}}_k = \boldsymbol{P}_k^-\boldsymbol{C}_k^T(\boldsymbol{C}_k\boldsymbol{P}_k^-\boldsymbol{C}_k^T + \boldsymbol{R}_k)^{-1}$;

// correction step

$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \boldsymbol{\mathcal{K}}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$;

$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{\mathcal{K}}_k\boldsymbol{C}_k)\boldsymbol{P}_k^-$;

return $\hat{\mathbf{x}}_k, \boldsymbol{P}_k$

Algorithm 2: Robot dual estimation with Kalman Filter

Data: $\hat{\mathbf{y}}_{k-1}$, $\boldsymbol{P}_{k-1}^y$, $\hat{\mathbf{p}}_{k-1}$, $\boldsymbol{P}_{k-1}^p$, $\mathbf{u}_{k-1,k}$, $\mathbf{m}_k$

// state estimation

$A^y = \boldsymbol{A}_y^*(\hat{\mathbf{p}}_{k-1})$;

$\mathbf{b}^y = \mathbf{b}_y^*(\hat{\mathbf{p}}_{k-1}, \mathbf{u}_{k-1,k})$;

$[\hat{\mathbf{y}}_k, \boldsymbol{P}_k^y] = \mathrm{KF}(\hat{\mathbf{y}}_{k-1}, \boldsymbol{P}_{k-1}^y, A^y, \boldsymbol{C}_y, \mathbf{m}_k, \boldsymbol{Q}_{k-1}^y, \boldsymbol{R}_k, \mathbf{b}^y, \mathbf{0})$; // KF

// parameter estimation

$C^p = \boldsymbol{C}_y^*(\hat{\mathbf{y}}_{k-1}, \mathbf{u}_{k-1})$;

$\mathbf{d}^p = \mathbf{d}_y^*(\hat{\mathbf{y}}_{k-1})$;

$R^p = \boldsymbol{C}_y\boldsymbol{Q}_{k-1}^y\boldsymbol{C}_y^T + \boldsymbol{R}_k$;

$[\hat{\mathbf{p}}_k, \boldsymbol{P}_k^p] = \mathrm{KF}(\hat{\mathbf{p}}_{k-1}, \boldsymbol{P}_{k-1}^p, \boldsymbol{I}, C^p, \mathbf{m}_k, \boldsymbol{Q}_{k-1}^p, R^p, \mathbf{0}, \mathbf{d}^p)$; // KF

return $\hat{\mathbf{y}}_k$, $\boldsymbol{P}_k^y$, $\hat{\mathbf{p}}_k$, $\boldsymbol{P}_k^p$

3.4 Droplet dual estimation using Unknown Input Kalman Filter

In this Section we develop a method based on Kalman filtering for the dual estimation of state and parameters of the droplet, without knowing the control input, i. e. the thermal gradient in droplet position.

Let us consider the dynamical model of thermocapillary motion of a droplet proposed in §2.4 and add dynamics of parameters and measurement equation, neglecting disturbances; for $t \in [0, T]$ it is:

$$\begin{aligned} \boldsymbol{M}_z(\mathbf{p}(t))\dot{\mathbf{z}}(t) &= \boldsymbol{A}_z(\mathbf{z}(t), \mathbf{p}(t))\, \mathbf{z}(t) + \boldsymbol{B}_z(\mathbf{z}(t), \mathbf{p}(t))\, \mathbf{d}(t) \\ \dot{\mathbf{p}}(t) &= 0 \\ \mathbf{m}(t) &= \boldsymbol{C}_z\, \mathbf{z}(t) \end{aligned} \tag{3.29}$$

where $\mathbf{z} \in \mathbb{R}^{n_z}$ is the droplet state vector, $\mathbf{p} \in \mathbb{R}^{n_p}$ its parameters, $\mathbf{d} \in \mathbb{R}^{n_d}$ the unknown control actions, $\mathbf{m} \in \mathbb{R}^{n_m}$ the measurement. The meaning of input matrix $\boldsymbol{B}_z$ depends on what the input is intended to be: if $\mathbf{d}$ is the vector of nodal temperatures derived from FE discretization, then input matrix depends on $\mathbf{z}$ and returns the thermal gradient in droplet position, multiplied by a coefficient. Otherwise, if unknown input $\mathbf{d}$ is the thermal gradient in droplet position, then $\boldsymbol{B}_z$ comprises only the coefficient in between thermal gradient and thermocapillary force. The latter case will be considered here, so $\boldsymbol{B}_z = \boldsymbol{B}_z(\mathbf{p})$, in order to get a simple procedure for droplet state estimation, without taking into account the whole thermal field of the liquid medium. Moreover, an outcome of this procedure is also an approximation of the temperature gradient in the actual droplet position, i. e. $\nabla_{\mathbf{x}} w(t, \mathbf{z}^x(t))$ for $t \in [0, T]$.

Time discretization Let us build a time grid $\Gamma = \{0 = t_0 < t_1 < \cdots < t_N = T\}$ and apply the forward difference scheme for time discretization, namely for $k = 1, 2, \ldots, N$:

$$\frac{\boldsymbol{M}_z(\mathbf{p}_k)\mathbf{z}_k - \boldsymbol{M}_z(\mathbf{p}_{k-1})\mathbf{z}_{k-1}}{\Delta t_k} = \boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\mathbf{z}_{k-1} + \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-1} \tag{3.30}$$

where the discrete-time state approximation is denoted $\mathbf{z}(t_k) \approx \mathbf{z}_k$ and $\Delta t_k = t_k - t_{k-1}$. Constant dynamics of parameters, $\mathbf{p}_k \approx \mathbf{p}_{k-1}$, leads to the following explicit scheme:

$$\begin{aligned} \boldsymbol{M}_z(\mathbf{p}_{k-1})\frac{\mathbf{z}_k - \mathbf{z}_{k-1}}{\Delta t_k} &= \boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\mathbf{z}_{k-1} + \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-1} \\ &\Downarrow \\ \mathbf{z}_k &= \mathbf{z}_{k-1} + \Delta t_k \boldsymbol{M}_z^{-1}(\mathbf{p}_{k-1}) \left[\boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\mathbf{z}_{k-1} + \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-1}\right] \\ &= \left[\boldsymbol{I} + \Delta t_k \boldsymbol{M}_z^{-1}(\mathbf{p}_{k-1})\boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\right]\mathbf{z}_{k-1} + \\ &\qquad + \left[\Delta t_k \boldsymbol{M}_z^{-1}(\mathbf{p}_{k-1})\boldsymbol{B}_z(\mathbf{p}_{k-1})\right]\mathbf{d}_{k-1} \end{aligned} \tag{3.31}$$

Defining function $\mathbf{f}_z$ and matrix $\boldsymbol{B}_z^*$ and adding process Gaussian independent noise $\boldsymbol{\nu}_k^z \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k^z)$ for $k \in \{0, 1, \ldots, N\}$, we get the non-linear time-varying discrete-time dynamics of droplet state $\mathbf{z}$. Nonlinearities are due to the quadratic drag force acting on the droplet; dynamics would be linear in presence of linear drag force. Update equation of parameters and measurement with noises $\boldsymbol{\nu}_k^p \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k^p)$ and $\boldsymbol{\nu}_k^m \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_k)$ are easily obtained. The stochastic discrete-time model of the system is:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{f}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1}) + \boldsymbol{B}_z^*(\mathbf{p}_{k-1})\mathbf{d}_{k-1} + \boldsymbol{\nu}_{k-1}^z \\ \mathbf{p}_k &= \mathbf{p}_{k-1} + \boldsymbol{\nu}_{k-1}^p \\ \mathbf{m}_k &= \boldsymbol{C}_z \mathbf{z}_k + \boldsymbol{\nu}_k^m \end{aligned} \tag{3.32}$$

Let us adopt now the central difference scheme for time discretization, as reported in §2.3.2.

$$\frac{\boldsymbol{M}_z(\mathbf{p}_k)\mathbf{z}_k - \boldsymbol{M}_z(\mathbf{p}_{k-1})\mathbf{z}_{k-1}}{\Delta t_k} = \frac{\boldsymbol{A}_z(\mathbf{z}_k, \mathbf{p}_k)\mathbf{z}_k + \boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\mathbf{z}_{k-1}}{2} + \\ + \frac{\boldsymbol{B}_z(\mathbf{p}_k)\mathbf{d}_k + \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-1}}{2} \tag{3.33}$$

Neglecting process noise in constant dynamics of droplet parameters and introducing the notation $\mathbf{d}_{k-\frac{1}{2}} = [\mathbf{d}_k + \mathbf{d}_{k-1}]/2$, the state update equation reduces to:

$$\boldsymbol{M}_z(\mathbf{p}_{k-1})\frac{\mathbf{z}_k - \mathbf{z}_{k-1}}{\Delta t_k} = \frac{\boldsymbol{A}_z(\mathbf{z}_k, \mathbf{p}_{k-1})\mathbf{z}_k + \boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\mathbf{z}_{k-1}}{2} + \\ + \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-\frac{1}{2}} \tag{3.34}$$

The additional assumption $\mathbf{z}_{k-1} \approx \mathbf{z}_k$, that is valid if droplet dynamics is slow enough, leads to an explicit update equation[8]:

$$\begin{gathered} \boldsymbol{M}_z(\mathbf{p}_{k-1})\frac{\mathbf{z}_k - \mathbf{z}_{k-1}}{\Delta t_k} = \boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\frac{\mathbf{z}_k + \mathbf{z}_{k-1}}{2} + \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-\frac{1}{2}} \\ \Downarrow \\ \left[\boldsymbol{M}_z(\mathbf{p}_{k-1}) - \frac{\Delta t_k}{2}\boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\right]\mathbf{z}_k = \\ = \left[\boldsymbol{M}_z(\mathbf{p}_{k-1}) + \frac{\Delta t_k}{2}\boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\right]\mathbf{z}_{k-1} + \Delta t_k \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-\frac{1}{2}} \end{gathered} \tag{3.35}$$

Finally, non-linear stochastic discrete-time update and measurement equations take the form:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{f}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1}) + \boldsymbol{B}_z^*(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\mathbf{d}_{k-\frac{1}{2}} + \boldsymbol{\nu}_{k-1}^z \\ \mathbf{p}_k &= \mathbf{p}_{k-1} + \boldsymbol{\nu}_{k-1}^p \\ \mathbf{m}_k &= \boldsymbol{C}_z\mathbf{z}_k + \boldsymbol{\nu}_k^m \end{aligned} \tag{3.36}$$

State estimation with unknown input Let us focus on droplet state estimation using model (3.36). For easy of notation let us write the model (for both forward and central difference scheme) as

$$\begin{aligned} \mathbf{z}_k &= \mathbf{f}_{k-1} + \boldsymbol{B}_{k-1}\mathbf{d}_{k-j} + \boldsymbol{\nu}_{k-1}^z \\ \mathbf{m}_k &= \boldsymbol{C}_z\mathbf{z}_k + \boldsymbol{\nu}_k^m \end{aligned} \tag{3.37}$$

where $\mathbf{f}_k = \mathbf{f}_z(\mathbf{z}_k, \mathbf{p}_k)$ for brevity; adopting the forward difference scheme it is $\boldsymbol{B}_k = \boldsymbol{B}_z^*(\mathbf{p}_k)$ and $j = 1$, and central difference corresponds to $\boldsymbol{B}_k = \boldsymbol{B}_z^*(\mathbf{z}_k, \mathbf{p}_k)$ and $j = 1/2$.

[8] Notice that a better approximation is given by the Heun's method. It computes an intermediate estimate of the next state $\hat{\mathbf{z}}_k$ given the previous $\mathbf{z}_{k-1}$ and using forward finite difference scheme. Then, estimation $\mathbf{z}_k$ is refined using trapezoidal rule with the first intermediate estimate $\hat{\mathbf{z}}_k$ instead of $\mathbf{z}_k$. In this way, the method is explicit and of second order.

$$\hat{\mathbf{z}}_k = \mathbf{z}_{k-1} + \Delta t_k \boldsymbol{M}_z^{-1}\left[\boldsymbol{A}_z(\mathbf{z}_{k-1})\mathbf{z}_{k-1} + \boldsymbol{B}_z\mathbf{d}_{k-1}\right]$$

The main problem is to design an estimator which is insensitive to the influence of the unknown input. The necessary condition for the existence of a solution to the unknown input problem is stated in [70]:

$$\operatorname{rank}\left(\boldsymbol{C}_z\boldsymbol{B}_{k-1}\right) = \operatorname{rank}\left(\boldsymbol{B}_{k-1}\right) = \min\left(n_z, n_d\right) \tag{3.38}$$

The second equality means that matrix $\boldsymbol{B}_{k-1}$ is full rank, being $\boldsymbol{B}_{k-1} \in \mathbb{R}^{n_z \times n_d}$. Substituting state update into measurement equation we can solve for the unknown input:

$$\begin{aligned} \mathbf{m}_k &= \boldsymbol{C}_z\mathbf{z}_k + \boldsymbol{\nu}_k^m \\ &= \boldsymbol{C}_z\left[\mathbf{f}_{k-1} + \boldsymbol{B}_{k-1}\mathbf{d}_{k-j} + \boldsymbol{\nu}_{k-1}^z\right] + \boldsymbol{\nu}_k^m \\ &\Downarrow \\ \boldsymbol{C}_z\boldsymbol{B}_{k-1}\mathbf{d}_{k-j} &= \mathbf{m}_k - \boldsymbol{C}_z\left(\mathbf{f}_{k-1} + \boldsymbol{\nu}_{k-1}^z\right) - \boldsymbol{\nu}_k^m \end{aligned} \tag{3.39}$$

If condition (3.38) holds, then the pseudo-inverse matrix $\boldsymbol{H}_k = \left(\boldsymbol{C}_z\boldsymbol{B}_{k-1}\right)^\dagger$ can be defined[9] and the unknown input is found:

$$\mathbf{d}_{k-j} = \boldsymbol{H}_k\left[\mathbf{m}_k - \boldsymbol{C}_z\left(\mathbf{f}_{k-1} + \boldsymbol{\nu}_{k-1}^z\right) - \boldsymbol{\nu}_k^m\right] \tag{3.40}$$

Let us substitute back into the update rule, obtaining the updated state as a function of measurement:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{f}_{k-1} + \boldsymbol{\nu}_{k-1}^z + \boldsymbol{B}_{k-1}\boldsymbol{H}_k\left[\mathbf{m}_k - \boldsymbol{C}_z\left(\mathbf{f}_{k-1} + \boldsymbol{\nu}_{k-1}^z\right) - \boldsymbol{\nu}_k^m\right] \\ &= \boldsymbol{G}_k\left(\mathbf{f}_{k-1} + \boldsymbol{\nu}_{k-1}^z\right) + \boldsymbol{B}_{k-1}\boldsymbol{H}_k\left(\mathbf{m}_k - \boldsymbol{\nu}_k^m\right) \end{aligned} \tag{3.41}$$

where matrix $\boldsymbol{G}_k = \boldsymbol{I} - \boldsymbol{B}_{k-1}\boldsymbol{H}_k\boldsymbol{C}_z$ is introduced.

At this point a Kalman filter may be applied for state estimation, with a minor modification in prediction step. In particular, given the previous state estimate $\hat{\mathbf{z}}_{k-1}$ with covariance matrix $\boldsymbol{P}_{k-1}^z$, the prediction step is given by

$$\begin{aligned} \hat{\mathbf{z}}_{k|k-1} &= \mathbb{E}\{\mathbf{z}_k\} = \hat{\boldsymbol{G}}_k\hat{\mathbf{f}}_{k-1} + \hat{\boldsymbol{B}}_{k-1}\hat{\boldsymbol{H}}_k\mathbf{m}_k \\ \boldsymbol{P}_{k|k-1}^z &= \operatorname{cov}\left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right) \\ &= \operatorname{cov}\left(\boldsymbol{G}_k\left(\mathbf{f}_{k-1} + \boldsymbol{\nu}_{k-1}^z\right) + \boldsymbol{B}_{k-1}\boldsymbol{H}_k\left(\mathbf{m}_k - \boldsymbol{\nu}_k^m\right) - \hat{\boldsymbol{G}}_k\hat{\mathbf{f}}_{k-1} - \hat{\boldsymbol{B}}_{k-1}\hat{\boldsymbol{H}}_k\mathbf{m}_k\right) \end{aligned} \tag{3.42}$$

Approximating actual and estimated matrices, i. e. $\hat{\boldsymbol{H}}_k \approx \boldsymbol{H}_k$, $\hat{\boldsymbol{G}}_k \approx \boldsymbol{G}_k$, and $\hat{\boldsymbol{B}}_{k-1} \approx \boldsymbol{B}_{k-1}$ and assuming independent noise, predicted covariance matrix is

$$\begin{aligned} \boldsymbol{P}_{k|k-1}^z &= \operatorname{cov}\left(\hat{\boldsymbol{G}}_k\left(\mathbf{f}_{k-1} - \hat{\mathbf{f}}_{k-1}\right) + \hat{\boldsymbol{G}}_k\boldsymbol{\nu}_{k-1}^z - \hat{\boldsymbol{B}}_{k-1}\hat{\boldsymbol{H}}_k\boldsymbol{\nu}_k^m\right) \\ &= \operatorname{cov}\left(\hat{\boldsymbol{G}}_k\left(\mathbf{f}_{k-1} - \hat{\mathbf{f}}_{k-1}\right)\right) + \hat{\boldsymbol{G}}_k\boldsymbol{Q}_{k-1}^z\hat{\boldsymbol{G}}_k^T + \hat{\boldsymbol{B}}_{k-1}\hat{\boldsymbol{H}}_k\boldsymbol{R}_k\hat{\boldsymbol{H}}_k^T\hat{\boldsymbol{B}}_{k-1}^T \end{aligned} \tag{3.43}$$

Then we linearise $\mathbf{f}_{k-1} \approx \hat{\mathbf{f}}_{k-1}$ around $\hat{\mathbf{z}}_{k-1}$, obtaining $\boldsymbol{A}_{k-1} \approx \hat{\boldsymbol{A}}_{k-1} = \nabla_{\mathbf{z}}\hat{\mathbf{f}}_{k-1}(\mathbf{z})\Big|_{\hat{\mathbf{z}}_{k-1}}$.
Hence the covariance matrix can be explicitly expressed as:

$$\begin{aligned} \boldsymbol{P}_{k|k-1}^z &= \operatorname{cov}\left(\hat{\boldsymbol{G}}_k\hat{\boldsymbol{A}}_{k-1}\left(\mathbf{z}_{k-1} - \hat{\mathbf{z}}_{k-1}\right)\right) + \hat{\boldsymbol{G}}_k\boldsymbol{Q}_{k-1}^z\hat{\boldsymbol{G}}_k^T + \hat{\boldsymbol{B}}_{k-1}\hat{\boldsymbol{H}}_k\boldsymbol{R}_k\hat{\boldsymbol{H}}_k^T\hat{\boldsymbol{B}}_{k-1}^T \\ &= \hat{\boldsymbol{G}}_k\hat{\boldsymbol{A}}_{k-1}\boldsymbol{P}_{k-1}\hat{\boldsymbol{A}}_{k-1}^T\hat{\boldsymbol{G}}_k^T + \hat{\boldsymbol{G}}_k\boldsymbol{Q}_{k-1}^z\hat{\boldsymbol{G}}_k^T + \hat{\boldsymbol{B}}_{k-1}\hat{\boldsymbol{H}}_k\boldsymbol{R}_k\hat{\boldsymbol{H}}_k^T \end{aligned} \tag{3.44}$$

[9] The left Moore-Penrose inverse of a matrix $\boldsymbol{M}$ is $\boldsymbol{M}^\dagger = \left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T$.

Then, traditional Kalman gain and subsequent correction step may be performed. Let us compute the so called innovation $\mathbf{s}$ and its covariance matrix $\boldsymbol{S}$, the Kalman gain $\boldsymbol{\mathcal{K}}$ and the state estimate correction:

$$\begin{aligned}
\mathbf{s}_k &= \mathbf{m}_k - \mathbb{E}\{\mathbf{m}_k\} = \mathbf{m}_k - \boldsymbol{C}_z\hat{\mathbf{z}}_{k|k-1} \\
\boldsymbol{S}_k &= \text{cov}\left(\mathbf{s}_k\right) = \text{cov}\left(\mathbf{m}_k - \boldsymbol{C}_z\hat{\mathbf{z}}_{k|k-1}\right) \\
&= \text{cov}\left(\boldsymbol{C}_z\left(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}\right) + \boldsymbol{\nu}_k^m\right) = \boldsymbol{C}_z\boldsymbol{P}_{k|k-1}^z\boldsymbol{C}_z^T + \boldsymbol{R}_k \\
\boldsymbol{\mathcal{K}}_k &= \boldsymbol{P}_{k|k-1}^z\boldsymbol{C}_z^T\boldsymbol{S}_k^{-1} \\
\hat{\mathbf{z}}_k &= \hat{\mathbf{z}}_{k|k-1} + \boldsymbol{\mathcal{K}}_k\mathbf{s}_k \\
\boldsymbol{P}_k^z &= (\boldsymbol{I} - \boldsymbol{\mathcal{K}}_k\boldsymbol{C}_z)\boldsymbol{P}_{k|k-1}^z
\end{aligned} \tag{3.45}$$

Finally, unknown input estimate and its covariance matrix can be computed from the actual state estimate, measurement and Eq. (3.40):

$$\begin{aligned}
\hat{\mathbf{d}}_{k-1} &= \mathbb{E}\{\mathbf{d}_{k-1}\} = \hat{\boldsymbol{H}}_k\left(\mathbf{m}_k - \boldsymbol{C}_z\hat{\mathbf{f}}_{k-1}\right) \\
\boldsymbol{P}_{k-1}^d &= \text{cov}\left(\mathbf{d}_{k-1} - \hat{\mathbf{d}}_{k-1}\right) \\
&= \text{cov}\left(\boldsymbol{H}_k\left[\mathbf{m}_k - \boldsymbol{C}_z\left(\mathbf{f}_{k-1} + \boldsymbol{\nu}_{k-1}^z\right) - \boldsymbol{\nu}_k^m\right] - \hat{\boldsymbol{H}}_k\left(\mathbf{m}_k - \boldsymbol{C}_z\hat{\mathbf{f}}_{k-1}\right)\right)
\end{aligned} \tag{3.46}$$

Assuming as before $\hat{\boldsymbol{H}}_k \approx \boldsymbol{H}_k$, $\mathbf{f}_{k-1} \approx \hat{\mathbf{f}}_{k-1}$ and the linear approximation of $\hat{\mathbf{f}}_{k-1}$ around $\hat{\mathbf{z}}_{k-1}$, the covariance matrix reduces to:

$$\begin{aligned}
\boldsymbol{P}_{k-1}^d &= \hat{\boldsymbol{H}}_k\text{cov}\left(\boldsymbol{C}_z\left[\mathbf{f}_{k-1} - \hat{\mathbf{f}}_{k-1}\right] + \boldsymbol{C}_z\boldsymbol{\nu}_{k-1}^z + \boldsymbol{\nu}_k^m\right)\hat{\boldsymbol{H}}_k^T \\
&= \hat{\boldsymbol{H}}_k\text{cov}\left(\boldsymbol{C}_z\hat{\boldsymbol{A}}_{k-1}\left(\mathbf{z}_{k-1} - \hat{\mathbf{z}}_{k-1}\right) + \boldsymbol{C}_z\boldsymbol{\nu}_{k-1}^z + \boldsymbol{\nu}_k^m\right)\hat{\boldsymbol{H}}_k^T \\
&= \hat{\boldsymbol{H}}_k\left(\boldsymbol{C}_z\hat{\boldsymbol{A}}_{k-1}\boldsymbol{P}_{k-1}^z\hat{\boldsymbol{A}}_{k-1}^T\boldsymbol{C}_z^T + \boldsymbol{C}_z\boldsymbol{Q}_{k-1}^z\boldsymbol{C}_z^T + \boldsymbol{R}_k\right)\hat{\boldsymbol{H}}_k^T
\end{aligned} \tag{3.47}$$

The procedure just derived for droplet state estimation with unknown input is a modified (extended) Kalman filter; it is schematically reported in Algorithm 3.

Parameters identification Let us focus on how to estimate droplet parameters $\mathbf{p} = [m_D, c_D, b_D]^T$. The procedure for online parameters identification proposed in §3.3 is here followed. Let us recall the discrete-time droplet dynamics (3.31), obtained by using explicit Euler method, and try to express the measurement equation as a function of parameters $\mathbf{p}$. With matrices presented in §2.4, the state update rule is:

$$\begin{aligned}
\mathbf{z}_k &= \mathbf{z}_{k-1} + \Delta t_k\boldsymbol{M}_z^{-1}(\mathbf{p}_{k-1})\left[\boldsymbol{A}_z(\mathbf{z}_{k-1}, \mathbf{p}_{k-1})\mathbf{z}_{k-1} + \boldsymbol{B}_z(\mathbf{p}_{k-1})\mathbf{d}_{k-1}\right] \\
&= \mathbf{z}_{k-1} + \Delta t_k\begin{bmatrix}\boldsymbol{I} & \\ & m_D^{-1}\boldsymbol{I}\end{bmatrix}\left(\begin{bmatrix}\mathbf{0} & \boldsymbol{I} \\ & -c_D\|\mathbf{z}_{k-1}^v\|\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\mathbf{z}_{k-1}^x \\ \mathbf{z}_{k-1}^v\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ b_D\boldsymbol{I}\end{bmatrix}\mathbf{d}_{k-1}\right) \\
&= \begin{bmatrix}\mathbf{z}_{k-1}^x + \Delta t_k\,\mathbf{z}_{k-1}^v \\ \mathbf{z}_{k-1}^v\end{bmatrix} + \Delta t_k\begin{bmatrix}\mathbf{0} & \\ -\|\mathbf{z}_{k-1}^v\|\mathbf{z}_{k-1}^v & \mathbf{d}_{k-1}\end{bmatrix}\begin{bmatrix}c_D/m_D \\ b_D/m_D\end{bmatrix}
\end{aligned} \tag{3.48}$$

Then, introducing suitable matrices $\boldsymbol{C}_p$ and $\boldsymbol{D}_p$ and defining vector $\tilde{\mathbf{p}} = [c_D/m_D, b_D/m_D]^T$, the measurement equation is given by:

$$\begin{aligned}
\mathbf{m}_k &= \boldsymbol{C}_z\mathbf{z}_k \\
&= \boldsymbol{C}_z\left[\boldsymbol{C}_p(\mathbf{z}_{k-1}, \mathbf{d}_{k-1})\tilde{\mathbf{p}}_{k-1} + \boldsymbol{D}_p(\mathbf{z}_{k-1})\right] \\
&= \boldsymbol{C}_z\boldsymbol{C}_p(\mathbf{z}_{k-1}, \mathbf{d}_{k-1})\tilde{\mathbf{p}}_k + \boldsymbol{C}_z\boldsymbol{D}_p(\mathbf{z}_{k-1})
\end{aligned} \tag{3.49}$$

Algorithm 3: Unknown-Input (Extended) Kalman Filter

Data: $\hat{\mathbf{x}}_{k-1}$, $\boldsymbol{P}_{k-1}$, $\mathbf{y}_k$

if *state transition is linear* **then**

  $\hat{\mathbf{x}}_{k-1}^{+} = \boldsymbol{A}_{k-1}\hat{\mathbf{x}}_{k-1}$;

else

  $\hat{\mathbf{x}}_{k-1}^{+} = f(\hat{\mathbf{x}}_{k-1})$;

  $\boldsymbol{A}_{k-1} = \nabla_{\mathbf{x}} f(\mathbf{x})\Big|_{\hat{\mathbf{x}}_{k-1}}$;

if $\operatorname{rank}(\boldsymbol{C}_k\boldsymbol{B}_{k-1}) \neq \operatorname{rank}(\boldsymbol{B}_{k-1})$ **then**

  return

$\boldsymbol{H}_k = (\boldsymbol{C}_k\boldsymbol{B}_{k-1})^{\dagger}$;

// input estimation

$\hat{\mathbf{u}}_{k-1} = \boldsymbol{H}_k\left(\mathbf{y}_k - \boldsymbol{C}_k\boldsymbol{A}_{k-1}\hat{\mathbf{x}}_{k-1}\right)$;

$\boldsymbol{P}_{k-1}^{u} = \boldsymbol{H}_k\left(\boldsymbol{C}_k\boldsymbol{A}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{A}_{k-1}^{T}\boldsymbol{C}_k^{T} + \boldsymbol{C}_k\boldsymbol{Q}_{k-1}\boldsymbol{C}_k^{T} + \boldsymbol{R}_k\right)\boldsymbol{H}_k^{T}$;

// prediction step

$\boldsymbol{G}_k = \boldsymbol{I} - \boldsymbol{B}_{k-1}\boldsymbol{H}_k\boldsymbol{C}_k$;

$\hat{\mathbf{x}}_{k|k-1} = \boldsymbol{G}_k\hat{\mathbf{x}}_{k-1}^{+} + \boldsymbol{B}_{k-1}\boldsymbol{H}_k\mathbf{y}_k$;

$\hat{\mathbf{y}}_k = \boldsymbol{C}_k\hat{\mathbf{x}}_{k|k-1}$;

$\boldsymbol{P}_{k|k-1} = \boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{P}_{k-1}\boldsymbol{A}_{k-1}^{T}\boldsymbol{G}_k^{T} + \boldsymbol{G}_k\boldsymbol{Q}_{k-1}\boldsymbol{G}_k^{T} + \boldsymbol{B}_{k-1}\boldsymbol{H}_k\boldsymbol{R}_k\boldsymbol{H}_k^{T}\boldsymbol{B}_{k-1}^{T}$;

// Kalman gain

$\boldsymbol{\mathcal{K}}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^{T}(\boldsymbol{C}_k\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^{T} + \boldsymbol{R}_k)^{-1}$;

// correction step

$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \boldsymbol{\mathcal{K}}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$;

$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{\mathcal{K}}_k\boldsymbol{C}_k)\boldsymbol{P}_{k|k-1}$;

return $\hat{\mathbf{x}}_k$, $\boldsymbol{P}_k$, $\hat{\mathbf{u}}_{k-1}$, $\boldsymbol{P}_{k-1}^{u}$

Hence, the stochastic discrete-time model of parameters dynamics is linear. This property does not hold is central difference scheme is adopted for time discretization, as shown in §3.3 for robot parameters dynamics.

$$\begin{aligned} \tilde{\mathbf{p}}_k &= \tilde{\mathbf{p}}_{k-1} + \boldsymbol{\nu}^p_{k-1} \\ \mathbf{m}_k &= \boldsymbol{C}_z \boldsymbol{C}_p(\mathbf{z}_{k-1}, \mathbf{d}_{k-1}) \tilde{\mathbf{p}}_k + \boldsymbol{C}_z \boldsymbol{D}_p(\mathbf{z}_{k-1}) + \boldsymbol{\nu}^m_k \end{aligned} \tag{3.50}$$

Thus, a linear Kalman filter may be applied to find step by step an estimate of $\tilde{\mathbf{p}}_k$, given previously estimated droplet state $\hat{\mathbf{z}}_{k-1}$ and input $\hat{\mathbf{d}}_{k-1}$.

Procedure A dual estimation procedure for droplet state and parameters is proposed in Algorithm 4, that integrates Algorithm 3 and a standard Kalman filter (Algorithm 1). It is worth to state again that uncertainties on estimates are not exchanged between the two filters, i. e. covariance matrices of state and input are not considered in parameter estimation, and *viceversa*.

Algorithm 4: Droplet dual estimation with Unknown Input Kalman Filter

Data: $\hat{\mathbf{z}}_{k-1}$, $\boldsymbol{P}_{k-1}$, $\hat{\mathbf{p}}_{k-1}$, $\boldsymbol{P}^p_{k-1}$, $\mathbf{m}_k$

// state and input estimation

$\mathbf{f}(\mathbf{z}) = \mathbf{f}_z(\mathbf{z}, \hat{\mathbf{p}}_{k-1})$;

$\boldsymbol{B} = \boldsymbol{B}^*_z(\hat{\mathbf{p}}_{k-1})$;

$[\hat{\mathbf{z}}_k, \boldsymbol{P}_k, \hat{\mathbf{d}}_{k-1}, \boldsymbol{P}^d_{k-1}] = \text{UI-EKF}\left(\hat{\mathbf{z}}_{k-1}, \boldsymbol{P}_{k-1}, \mathbf{f}, \boldsymbol{B}, \boldsymbol{C}_z, \mathbf{m}_k, \boldsymbol{Q}^z, \boldsymbol{R}\right)$; // UI-EKF

// parameter estimation

$\boldsymbol{C} = \boldsymbol{C}_z \boldsymbol{C}_p(\hat{\mathbf{z}}_{k-1}, \hat{\mathbf{d}}_{k-1})$;

$\boldsymbol{D} = \boldsymbol{C}_z \boldsymbol{D}_p(\hat{\mathbf{z}}_{k-1})$;

$[\hat{\mathbf{p}}_k, \boldsymbol{P}^p_k] = \text{KF}\left(\hat{\mathbf{p}}_{k-1}, \boldsymbol{P}^p_{k-1}, \boldsymbol{I}, \boldsymbol{C}, \mathbf{m}_k, \boldsymbol{Q}^p, \boldsymbol{R}, \mathbf{0}, \boldsymbol{D}\right)$; // KF

return $\hat{\mathbf{z}}_k$, $\boldsymbol{P}_k$, $\hat{\mathbf{d}}_{k-1}$, $\boldsymbol{P}^d_{k-1}$, $\hat{\mathbf{p}}_k$, $\boldsymbol{P}^p_k$

3.5 Liquid dual estimation using Kalman Filter

This Section focuses on the dual estimation problem for the liquid medium. The objective is to develop a feasible and efficient algorithm to estimate state and parameters of the liquid layer as described in the model proposed in §2.3.2.

Let us consider the dynamical model based on FE discretization and add dynamics of parameters and measurement equation, neglecting disturbances; for $t \in [0, T]$ it is:

$$\begin{aligned} \boldsymbol{M}_w(\mathbf{p}(t))\dot{\mathbf{w}}(t) &= -\boldsymbol{K}_w(\mathbf{p}(t))\,\mathbf{w}(t) + \boldsymbol{A}_{wu}(\mathbf{y}(t), \mathbf{p}(t))\,\mathbf{u}(t) \\ \dot{\mathbf{p}}(t) &= 0 \\ \mathbf{m}(t) &= \boldsymbol{C}_w(\mathbf{z}(t))\,\mathbf{w}(t) \end{aligned} \tag{3.51}$$

where $\mathbf{w}$ is the nodal vector, i. e. thermal state of the liquid, $\mathbf{p}$ the liquid parameters, $\mathbf{u}$ the control input, $\mathbf{m}$ the measurement, $\mathbf{y}$ the robot state and $\mathbf{z}$ the droplet state. Notice that measurement $\mathbf{m}$ may be a subset of the nodal vector[10] or it may correspond to thermal gradient in droplet position estimated during the droplet dual estimation.

[10] A FE–based Kalman filter has been developed to deal with spatially distributed problems and a finite set of sampling points. See [71–73] for details.

In the latter case, matrix $\boldsymbol{C}_w(\mathbf{z})$ may represent both an exact computation of thermal gradient based on shape functions or an approximation based on finite differences[11].

Time discretization Let us build a time grid $\Gamma = \{0 = t_0 < t_1 < \cdots < t_N = T\}$ and apply the forward difference scheme for time discretization, namely for $k = 1, 2, \ldots, N$:

$$\frac{\boldsymbol{M}_w(\mathbf{p}_k)\mathbf{w}_k - \boldsymbol{M}_w(\mathbf{p}_{k-1})\mathbf{w}_{k-1}}{\Delta t_k} = -\boldsymbol{K}_w(\mathbf{p}_{k-1})\mathbf{w}_{k-1} + \boldsymbol{A}_{wu}(\mathbf{y}_{k-1}, \mathbf{p}_{k-1})\mathbf{u}_{k-1} \tag{3.52}$$

where $\mathbf{w}(t_k) \approx \mathbf{w}_k$ denotes the state approximation and $\Delta t_k = t_k - t_{k-1}$ the time step size. Assuming $\mathbf{p}_k \approx \mathbf{p}_{k-1}$, i. e. constant or slowly-varying parameters, the state update equation reduces to:

$$\begin{aligned} \boldsymbol{M}_w(\mathbf{p}_{k-1})\frac{\mathbf{w}_k - \mathbf{w}_{k-1}}{\Delta t_k} &= -\boldsymbol{K}_w(\mathbf{p}_{k-1})\mathbf{w}_{k-1} + \boldsymbol{A}_{wu}(\mathbf{y}_{k-1}, \mathbf{p}_{k-1})\mathbf{u}_{k-1} \\ &\Downarrow \\ \boldsymbol{M}_w(\mathbf{p}_{k-1})\mathbf{w}_k &= \left[\boldsymbol{M}_w(\mathbf{p}_{k-1}) - \Delta t_k \boldsymbol{K}_w(\mathbf{p}_{k-1})\right]\mathbf{w}_{k-1} + \\ &\quad + \Delta t_k \boldsymbol{A}_{wu}(\mathbf{y}_{k-1}, \mathbf{p}_{k-1})\mathbf{u}_{k-1} \end{aligned} \tag{3.53}$$

Then, let us adopt the central difference scheme for time discretization, as shown in §2.3.2, that leads to:

$$\begin{aligned} \frac{\boldsymbol{M}_w(\mathbf{p}_k)\mathbf{w}_k - \boldsymbol{M}_w(\mathbf{p}_{k-1})\mathbf{w}_{k-1}}{\Delta t_k} &= -\frac{\boldsymbol{K}_w(\mathbf{p}_k)\mathbf{w}_k + \boldsymbol{K}_w(\mathbf{p}_{k-1})\mathbf{w}_{k-1}}{2} + \\ &\quad + \frac{\boldsymbol{A}_{wu}(\mathbf{y}_k, \mathbf{p}_k)\mathbf{u}_k + \boldsymbol{A}_{wu}(\mathbf{y}_{k-1}, \mathbf{p}_{k-1})\mathbf{u}_{k-1}}{2} \end{aligned} \tag{3.54}$$

Considering $\mathbf{p}_k \approx \mathbf{p}_{k-1}$ as above, an explicit update equation of nodal vector $\mathbf{w}$ is obtained.

$$\begin{aligned} \left[\boldsymbol{M}_w(\mathbf{p}_{k-1}) + \frac{\Delta t_k}{2}\boldsymbol{K}_w(\mathbf{p}_{k-1})\right]\mathbf{w}_k &= \left[\boldsymbol{M}_w(\mathbf{p}_{k-1}) - \frac{\Delta t_k}{2}\boldsymbol{K}_w(\mathbf{p}_{k-1})\right]\mathbf{w}_{k-1} + \\ &\quad + \Delta t_k \frac{\boldsymbol{A}_{wu}(\mathbf{y}_k, \mathbf{p}_{k-1})\mathbf{u}_k + \boldsymbol{A}_{wu}(\mathbf{y}_{k-1}, \mathbf{p}_{k-1})\mathbf{u}_{k-1}}{2} \end{aligned} \tag{3.55}$$

With both discretization schemes, collecting terms into $\boldsymbol{A}_w^*$ and $\mathbf{b}_w^*$ and adding process Gaussian independent noise $\boldsymbol{\nu}_k^w \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k^w)$ for $k \in \{0, 1, \ldots, N\}$, a linear time-varying discrete-time dynamics of liquid thermal state $\mathbf{w}$ is found. Update equation of parameters and measurement with noises $\boldsymbol{\nu}_k^p \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k^p)$ and $\boldsymbol{\nu}_k^m \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_k)$ are easily obtained. Control input $\mathbf{u}_k$ is considered to be known without uncertainty for every $k \in \{0, 1, \ldots, N\}$. The stochastic discrete-time thermal model of the liquid layer is:

$$\begin{aligned} \mathbf{w}_k &= \boldsymbol{A}_w^*(\mathbf{p}_{k-1})\mathbf{w}_{k-1} + \mathbf{b}_w^*(\mathbf{y}_{k,k-1}, \mathbf{p}_{k-1}, \mathbf{u}_{k,k-1}) + \boldsymbol{\nu}_{k-1}^w \\ \mathbf{p}_k &= \mathbf{p}_{k-1} + \boldsymbol{\nu}_{k-1}^p \\ \mathbf{m}_k &= \boldsymbol{C}_w(\mathbf{z}_k)\mathbf{w}_k + \boldsymbol{\nu}_k^m \end{aligned} \tag{3.56}$$

Dynamics and measurement equation are linearly dependent on state, thus liquid state $\mathbf{w}$ could be estimated using the standard linear Kalman filter (Algorithm 1).

[11] Temperature gradient can be approximated using FEM by $\nabla_{\mathbf{x}} w(t, \mathbf{x}) \approx \boldsymbol{B}(\mathbf{x})\mathbf{w}(t)$ where $\boldsymbol{B}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathbf{n}(\mathbf{x})$ and $\mathbf{n}$ collects the shape functions of the FE discretization. Alternatively, one could choose to adopt the finite difference approximation $\nabla_{\mathbf{x}}^h \mathbf{n}(\mathbf{x})$ of $\nabla_{\mathbf{x}}\mathbf{n}(\mathbf{x})$, that returns $\nabla_{\mathbf{x}} w(t, \mathbf{x}) \approx \nabla_{\mathbf{x}}^h \mathbf{n}(\mathbf{x})\,\mathbf{w}(t)$.

Parameters identification Let us try to express measurement equation in terms of parameters $\mathbf{p}$, considering the forward difference scheme for time discretization. It is worth to extract (constant homogeneous) parameters from matrices[12]:

$$\boldsymbol{M}_w(\mathbf{p}) = \bar{\boldsymbol{M}}_w l_z c_p \varrho \ , \qquad \boldsymbol{K}_w(\mathbf{p}) = \bar{\boldsymbol{K}}_w l_z k \ , \qquad \boldsymbol{A}_{wu}(\mathbf{y}, \mathbf{p}) = \bar{\boldsymbol{A}}_{wu}(\mathbf{y})\beta \tag{3.57}$$

From (3.53) the updated state can be computed

$$\mathbf{w}_k = \left(\boldsymbol{I} - \Delta t_k \alpha \bar{\boldsymbol{M}}_w^{-1} \bar{\boldsymbol{K}}_w\right) \mathbf{w}_{k-1} + \Delta t_k \frac{\beta}{l_z c_p \varrho} \bar{\boldsymbol{M}}_w^{-1} \bar{\boldsymbol{A}}_{wu}(\mathbf{y}_{k-1}) \mathbf{u}_{k-1} \tag{3.58}$$

Then, defining vector $\tilde{\mathbf{p}} = [\tilde{p}^\alpha, \tilde{p}^\beta]^T = [\alpha, \beta/(l_z c_p \varrho)]^T$, state equation and measurement equation are linearly dependent on $\tilde{\mathbf{p}}$. As shown in §3.3 for robot parameters, this does not hold if the central difference scheme is adopted for time discretization.

$$\begin{aligned} \mathbf{m}_k &= \boldsymbol{C}_w(\mathbf{z}_k)\mathbf{w}_k \\ &= \boldsymbol{C}_w(\mathbf{z}_k)\left(\mathbf{w}_{k-1} - \Delta t_k \bar{\boldsymbol{M}}_w^{-1} \bar{\boldsymbol{K}}_w \mathbf{w}_{k-1} \tilde{p}^\alpha + \Delta t_k \bar{\boldsymbol{M}}_w^{-1} \bar{\boldsymbol{A}}_{wu}(\mathbf{y}_{k-1}) \mathbf{u}_{k-1} \tilde{p}^\beta\right) \\ &= \boldsymbol{C}_w(\mathbf{z}_k)\boldsymbol{C}_p(\mathbf{w}_{k-1}, \mathbf{y}_{k-1}, \mathbf{u}_{k-1})\tilde{\mathbf{p}}_k + \boldsymbol{C}_w(\mathbf{z}_k)\mathbf{w}_{k-1} \end{aligned} \tag{3.59}$$

Procedure Algorithm 5 resumes the procedure proposed for liquid dual estimation based on (linear) Kalman filtering. Notice that robot position $\hat{\mathbf{y}}$ and droplet position $\hat{\mathbf{z}}$ are employed without considering their uncertainties. Embedding these uncertainties would require to evaluate at each time step a linear approximation of $\bar{\boldsymbol{A}}_{wu}(\mathbf{y})$ around $\hat{\mathbf{y}}_k$ and $\boldsymbol{C}_w(\mathbf{z})$ around $\hat{\mathbf{z}}_k$. Further analysis are needed to discuss the impact of this choice on the accuracy and computational cost of the solution.

Due to the FE discretization, the implementation of this algorithm has to be analysed in details, because it may suffer the high dimensionality of vector $\mathbf{w}$. In particular, the computation of predicted covariance matrix $\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T$ may be an issue. Matrix multiplication algorithms take time on the order of n^ω with $\omega \in [2, 3]$ to multiply two $n \times n$ matrices. Considering that nodal vector $\mathbf{w}$ has $n \geq 500$ states, depending on mesh refinement, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\boldsymbol{P}_{k-1} \in \mathbb{R}^{n \times n}$ and computation of $\boldsymbol{P}_k^-$ takes time on the order of $n^{2\omega}$, then it is likely a bottleneck in the execution of the algorithm.

Algorithm 5: Liquid dual estimation using Kalman Filter

Data: $\hat{\mathbf{w}}_{k-1}$, $\boldsymbol{P}_{k-1}$, $\hat{\mathbf{p}}_{k-1}$, $\boldsymbol{P}_{k-1}^p$, $\mathbf{m}_k$, $\mathbf{u}_{k-1}$, $\hat{\mathbf{y}}_{k-1}$, $\hat{\mathbf{z}}_k$

// state estimation

$A = \boldsymbol{A}_w^*(\hat{\mathbf{p}}_{k-1})$;

$\mathbf{b} = \mathbf{b}_w^*(\hat{\mathbf{y}}_{k-1}, \hat{\mathbf{p}}_{k-1}, \mathbf{u}_{k-1})$;

$C = \boldsymbol{C}_w(\hat{\mathbf{z}}_k)$;

$[\hat{\mathbf{w}}_k, \boldsymbol{P}_k] = \mathrm{KF}\left(\hat{\mathbf{w}}_{k-1}, \boldsymbol{P}_{k-1}, A, C, \mathbf{m}_k, \boldsymbol{Q}_{k-1}^w, \boldsymbol{R}_k, \mathbf{b}, \mathbf{0}\right)$; // KF

// parameter estimation

$C^p = \boldsymbol{C}_w(\hat{\mathbf{z}}_k)\boldsymbol{C}_p(\hat{\mathbf{w}}_{k-1}, \hat{\mathbf{y}}_{k-1}, \mathbf{u}_{k-1})$;

$\mathbf{d} = \boldsymbol{C}_w(\hat{\mathbf{z}}_k)\hat{\mathbf{w}}_{k-1}$;

$[\hat{\mathbf{p}}_k, \boldsymbol{P}_k^p] = \mathrm{KF}\left(\hat{\mathbf{p}}_{k-1}, \boldsymbol{P}_{k-1}^p, \boldsymbol{I}, C^p, \mathbf{m}_k, \boldsymbol{Q}_{k-1}^p, \boldsymbol{R}_k, \mathbf{0}, \mathbf{d}\right)$; // KF

return $\hat{\mathbf{w}}_k$, $\boldsymbol{P}_k$, $\hat{\mathbf{p}}_k$, $\boldsymbol{P}_k^p$

[12]Notice that laser characteristic radius r_L cannot be explicit because of the exponential term. For the sake of simplicity, we discard r_L for the online parameters identification.

The approach presented in this Section for liquid state estimation may be quite costly from the computational point of view. Different strategies should be developed and tested, in order to gain real-time capabilities and accurate estimation. A thermocamera or a set of thermocouples would allow to build a distributed Kalman Filter [73] or a FE Kalman Filter [71, 72]. Another approach is Moving Horizon Estimation (MHE), that is a computationally feasible strategy for state estimation based on non-linear constrained optimization [66]. It has been noticed that non-linear MPC and MHE share the same optimal control structure and are often cited as "dual" problems [67]. Somehow the optimization-based approach adopted in §3.2 is in the more general and developed framework of MHE.

Chapter 4

Optimal control

This Chapter deals with the automatic control of the system in order to obtain a desired goal. In particular, optimal control theory is exploited and applied to design a model-based controller able to predict the system evolution and then to plan the optimal control input. The main idea is to use models developed in Chapter 2 with parameters estimated in Chapter 3 for calculating the optimum time-sequence of controls that ensure the robot moves the droplet as desired. As the real process does not coincide with the mathematical model and is most probably subject to disturbances, a feedback loop has to be established with an optimization-based controller.

Optimal control requires to define a cost functional to be minimized (or a reward functional to be maximized) by the optimal sequence of control inputs. This functional is a measure of the control optimality from the user point-of-view. In a pure time-optimal control problem, the fastest trajectory is the best one, but it may be not in a minimum-energy control problem, for instance. Optimal control problems with different cost functional correspond to different user objectives and have different solutions. Minimum time, minimum energy, reference tracking, target following, obstacle avoidance and robust control are examples of well-known control problems.

In [74] we proposed a control strategy for the time-optimal control problem with constraints on the final state. The suggested control scheme is reported in Fig. 4.1. Given a nominal path and the model of the droplet dynamics, one can calculate the optimal time-sequence of laser positions that would drive the droplet from its initial position to the final one (MPC). Then the sequence of robot controls is obtained thanks to the model of robot dynamics. At this step, a guess solution, i. e. an approximate sub-optimal solution, is available and it is possibly used for initializing the numerical routine for solving the time-optimal control problem (OCP). The result is a sequence of robot controls that allows to move the droplet from its initial state to the end of the nominal path, minimizing the user-specified cost and complying with constraints on states and controls.

In this work we enlarge the explored area adopting several techniques to solve slightly different problems, e. g. fixed final state, tracking and following OCPs.

4.1 Problem formulation

This Section states the minimum time OCP, the reference tracking OCP and the target following OCP. Model-based controllers are designed to find a solution of a mathematical optimization problem composed by an objective functional and constraints.

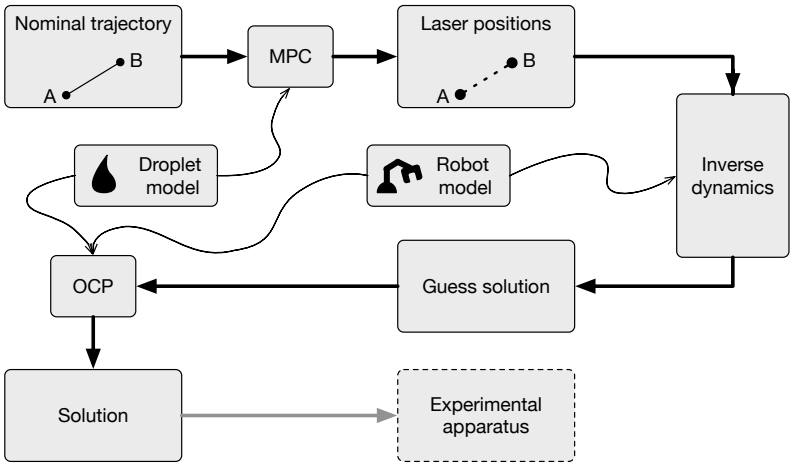


Figure 4.1: Schematics of control method proposed in [74].

Let us denote state vector $\mathbf{x} \in \mathbb{R}^{n_x}$ and control vector $\mathbf{u} \in \mathbb{R}^{n_u}$, so that $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are state of the system and control input at time $t \in [t_i,\, t_f]$. Let us consider w.l.o.g. the time interval $[t_i,\, t_f] = [0,\, T] \subset \mathbb{R}$. State and control are typically constrained in a subset, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{n_u}$. Vector $\mathbf{x}$ may be composed by the states of robot, liquid and droplet, i. e. $\mathbf{x} = [\mathbf{y}, \mathbf{w}, \mathbf{z}]^T$; alternatively, $\mathbf{x}$ may be a function of both time and space if liquid domain is not discretized in space.

Minimum time Let us define the cost functional $\mathcal{J}$ for the pure time-optimal control problem. Typically an additional cost on control magnitude is considered in order to limit the control effort. It is clearly seen that no reference paths or the like are given, so this type of problem is typically referred to as trajectory optimization.

$$\mathcal{J} = T = \int_0^T \mathrm{d}t \tag{4.1}$$

Collecting dynamics of robot, liquid layer and droplet, the continuous-time dynamic model and boundary conditions in general form can be written for $t \in [0, T]$, $\mathbf{x} \in \mathcal{X}$, $\mathbf{u} \in \mathcal{U}$ as:

$$\dot{\mathbf{x}}(t) = \boldsymbol{\varphi}(t, \mathbf{x}(t), \mathbf{u}(t)) \qquad\qquad \mathbf{b}(\mathbf{x}(0), \mathbf{x}(T)) = \mathbf{0} \tag{4.2}$$

Let us focus on a specific type of conditions: initial state is fixed and final state is partially or fully constrained, namely $\mathbf{x}(0) = \mathbf{x}^i$ and $\boldsymbol{B}_T \mathbf{x}(T) = \mathbf{x}_B^f$. The subset of constrained states at final time $t = T$ is selected by matrix $\boldsymbol{B}_T$. Initial state corresponds to the condition of the system at time $t = 0$, i. e. initial condition is not an optimization variable but a parameter of the optimization problem. At least one state in final condition has to be constrained, otherwise the problem has a trivial and useless solution — that is $\mathcal{J} = T = 0$ and $\mathbf{x}(T) = \mathbf{x}(0) = \mathbf{x}^i$. Final position of the droplet may be the objective of the control system and is typically fixed. Instead, droplet final velocity may be fixed or not, depending on the purpose.

Constraints on states and controls are often present and in general these require to satisfy a set of non-linear inequalities $\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0}$ for $t \in [0, T]$. For the sake of simplicity we decouple states and controls and consider simple constant bounds on each

of them, namely $\mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u$ and $\mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u$ $\forall t \in [0, T]$, where inequalities hold component-wise.

It is worth noticing that imposed constraints are hard constraints, i. e. they set conditions on variables that are required to be satisfied. One may impose soft constraints, that penalize values of variables in the objective function, adding Mayer terms for boundary conditions and Lagrange terms for state and control bounds. Typically these terms are in the form of penalty or barrier, depending on tolerance and flexibility of the constraints, and affect the convergence of the numerical solution.

The optimal control problem, called TOCP, consists in solving the following:

$$\begin{aligned} \text{minimize} \quad & T \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = \boldsymbol{\varphi}(t, \mathbf{x}(t), \mathbf{u}(t)) \\ & \mathbf{x}(0) = \mathbf{x}^i \\ & \boldsymbol{B}_T \mathbf{x}(T) = \mathbf{x}_B^f \\ & \mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u \\ & \mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u \\ \text{for} \quad & t \in (0, T) \end{aligned} \tag{4.3}$$

Objective functional and final condition constraint depend on final time T, that is *a priori* an unknown; hence (4.3) is a free final time problem and T is an optimization variable.

Reference tracking Let us focus on the reference tracking, i. e. the problem of changing the system from one state to another as fast as possible and following a given trajectory in the state space as close as possible. In particular, a droplet has to move from its actual position along a nominal user-specified path, as fast and accurately as possible. Curvilinear coordinates are introduced to ease the formulation of a suitable cost functional, as inspired by [75].

Given a nominal path γ curvilinear coordinates can be introduced to describe the droplet state (position and velocity) as projected on γ. Their representation is derived and reported in §4.4. Let us consider the dynamic model of the system and the kinematics of curvilinear coordinates $\boldsymbol{\xi}$, for a given final time T, $t \in [0, T]$, $\mathbf{x} \in \mathcal{X}$, $\mathbf{u} \in \mathcal{U}$, and initial conditions:

$$\dot{\mathbf{x}}(t) = \boldsymbol{\varphi}(t, \mathbf{x}(t), \mathbf{u}(t)) \qquad \mathbf{x}(0) = \mathbf{x}^i \tag{4.4}$$

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{R}_\gamma(\boldsymbol{\xi}(t))\, \mathbf{x}(t) \qquad \boldsymbol{\xi}(0) = \boldsymbol{\xi}^i \tag{4.5}$$

Matrix-valued function $\boldsymbol{R}_\gamma$ maps the absolute velocity of the droplet into velocities along the tangential and normal directions of γ, in the point of γ that is the closest to the actual position of the droplet. Initial conditions are known (perhaps approximately) and imposed, while state and control constraints $\mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u$ and $\mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u$ are typically enforced $\forall t \in [0, T]$.

Let us define a cost functional that takes into account both path tracking and minimum time features. It turns out that this definition is quite straightforward with curvilinear coordinates. In particular, in order to avoid a free final time problem, minimum time requirement is expressed as maximum displacement along the path in a given time interval. Then, tracking accuracy means to minimize the motion in the normal direction of the curve γ. A Mayer term $\mathcal{M}(\boldsymbol{\xi}(T))$ may reward the final position along γ and the Lagrange term $l(\boldsymbol{\xi})$ penalizes deviations from it. Curvilinear

coordinates collect curvilinear abscissa $s \geq 0$ and lateral deviation n, namely $\boldsymbol{\xi} = [s, n]^T$. Thus a suitable cost functional for the reference tracking problem may be $\mathcal{J}_{\text{MPC}}$, if control cost is neglected.

$$\mathcal{J}_{\text{MPC}} = \mathcal{M}(\boldsymbol{\xi}(T)) + \int_0^T l(\boldsymbol{\xi}(t))\,\mathrm{d}t = -s(T) + \omega_n \int_0^T n^2(t)\,\mathrm{d}t \tag{4.6}$$

Notice that the two terms have opposite signs, because the final abscissa $s(T)$ has to be maximized while the mean quadratic deviation has to be minimized. Moreover, notice the linear reward for distance $s(T)$ and the quadratic running cost for the lateral deviation. The relative importance of these two terms is ruled by the user-specified parameter ω_n. By setting $\omega_n = 0$ the contribution of lateral deviation vanishes, whereas for $\omega_n \to +\infty$ the tracking accuracy gains more importance than tracking speed.

The reference tracking optimal control problem, called TMPC, consists in solving the following:

$$\begin{aligned}
\text{minimize} \quad & \mathcal{J}_{\text{MPC}} \\
\text{subject to} \quad & \dot{\mathbf{x}}(t) = \boldsymbol{\varphi}(t, \mathbf{x}(t), \mathbf{u}(t)) \\
& \dot{\boldsymbol{\xi}}(t) = \boldsymbol{R}_\gamma(\boldsymbol{\xi}(t))\,\mathbf{x}(t) \\
& \mathbf{x}(0) = \mathbf{x}^i \\
& \boldsymbol{\xi}(0) = \boldsymbol{\xi}^i \\
& \mathbf{x}_l \leq \mathbf{x}(t) \leq \mathbf{x}_u \\
& \mathbf{u}_l \leq \mathbf{u}(t) \leq \mathbf{u}_u \\
\text{for} \quad & t \in (0, T)
\end{aligned} \tag{4.7}$$

Problem (4.7) refers to a tracking control with no final point and therefore it is well suited for unlimited—long enough—paths. It may be desirable to move the droplet to the end of curve γ, i. e. at $s = L \in (0, +\infty)$. In this case a suitable cost functional for the reference tracking problem may be (4.8), thanks to the quadratic cost on the final error $|s(T) - L|$. While minimizing lateral deviations, this cost functional tends to move the droplet at the final point of the reference path, and to keep it there.

$$(s(T) - L)^2 + \omega_n \int_0^T n^2(t)\,\mathrm{d}t \tag{4.8}$$

$$\log\cosh\left(s(T) - L\right) + \omega_n \int_0^T n^2(t)\,\mathrm{d}t \tag{4.9}$$

Instead of using the quadratic cost, pseudo-Huber loss function or log cosh function may be adopted to define the cost functional [37], especially for the Mayer term. The alternative definition (4.9) leads to asymptotically linear cost when $|s - L| \to +\infty$ and becomes asymptotically quadratic cost at $s \approx L$. This means that (4.9) converges to $\mathcal{J}_{\text{MPC}}$ as $s \to -\infty$. Moreover, far from the target, derivatives of $\mathcal{J}_{\text{MPC}}$ and (4.9) w.r.t. curvilinear abscissa s do not depend on the actual position. Thus, the optimal control is not influenced by the distance $|s - L|$ and assume a steady state value (neglecting lateral deviation and sign of $s - L$). Instead, close to the target, $s \approx L$, cost (4.9) is approximately quadratic w.r.t. $s(T)$ and then the optimal control depends on the actual position of the droplet.

4.2 Methods

Different approaches are possible to face optimal control problems formulated in §4.1. These problems are in general non-linear and most probably do not have an analytic solution. A special case is the exception of linear-quadratic OCPs. Hence, it is necessary to adopt numerical methods to solve these optimization problems. Two main approaches exist to deal with OCPs: indirect and direct methods. Indirect methods employ the calculus of variations to obtain the necessary first-order optimality conditions, that typically result in a two-point boundary value problem (BVP). Direct methods parametrize state and/or control using suitable function approximations and correspondingly the cost functional turns into a cost function. Then the infinite-dimensional problem is converted into a finite-dimensional problem where coefficients of approximations are optimization variables. A survey on well-known approaches is reported in [76] and some applications are [3, 5, 7, 27, 28, 35, 40, 75].

Original OCPs formulated in §4.1 involve a mix of ODEs and PDEs due to the thermal model of the liquid medium and mechanical model of robot and droplet. Expressing or approximating the dynamic model in different domains, for instance by using the FEM, and depending on the favoured approach, many different strategies are generated to solve the original OCPs. Possible ways to face the problem are graphically sketched in Fig. 4.2.

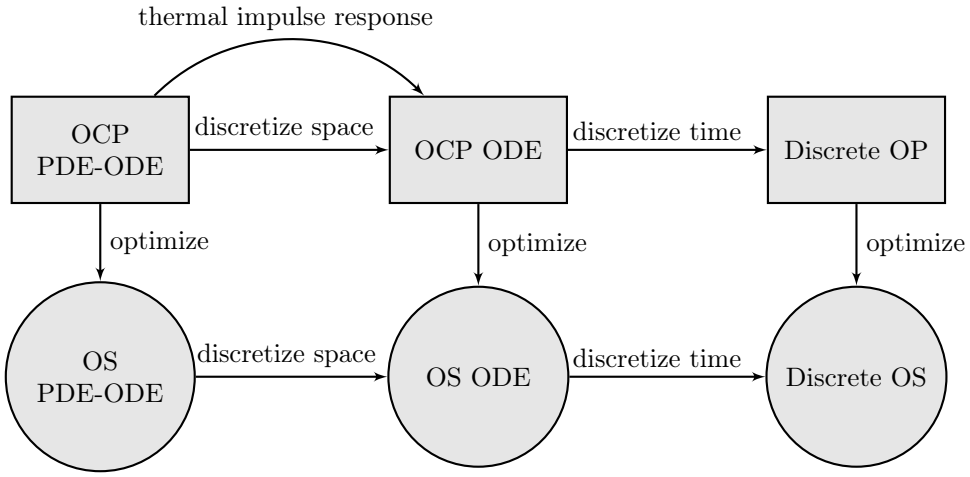


Figure 4.2: Flow chart of possible strategies to face OCPs.

Problems in §4.1 represent the starting point, OCP PDE-ODE. On the other side, non-linear optimization problems (NLP), called Discrete OP in Fig. 4.2, can be handled by state-of-art optimizers, e. g. IPOPT and WORHP. Non-linear solvers like CoDoSol can deal with systems of equations, Discrete OS, derived by necessary optimality conditions. Solvers of non-linear two-points BVPs could find a solution to OS ODE, that is an optimality system involving ODEs.

Feasible paths in Fig. 4.2 from the original problem to an optimality system correspond to different possible approaches. Four strategies can be univocally identified by the step at which optimization happens. In the following we will refer to them as *indirect*, *time-indirect*, *direct* and *analytical*, respectively for the optimization from OCP PDE-ODE, OCP ODE, Discrete OP and Discrete OP with an analytical solution for

the temperature field.

Indirect strategy This approach is inspired by the Pontryagin's Maximum Principle [77]. First-order necessary optimality conditions are derived and a BVP that involves PDEs and ODEs is obtained. Finite element method and standard numerical schemes (e. g. Crank-Nicholson scheme) may be used to discretize spatial domain and time interval; the result is typically a large-scale non-linear system, structured and sparse because of the FE parametrization [6, 23]. Then it is possible to find numerically an approximate solution by using a non-linear solver.

Time-indirect strategy Spatially-distributed dynamics may be parametrized using FEM, yielding to a dynamic model described by a system of ODEs. First-order necessary optimality conditions turn into a two-points BVP. Alternatively, function parametrization using finite difference schemes yields to a non-linear system of equations, that is structured due to the discretization of dynamic constraints. State-of-art non-linear solvers may find a solution to this type of problems. This approach has been applied to the optimal control of flexible links or robots [29, 30, 33].

Direct strategy Cost functional and constraints are discretized both in space and time domain, transforming the original control problem into a finite dimensional optimization problem [6]. Standard or *ad hoc* optimizers may take advantage of the particular structure of the problem for approximating the solution. This approach has risen major importance in the last decades thanks to the higher computational power available and developments in reduced-order modelling [27, 28, 31]. Notice that it is not always needed to convert explicitly the problem into a non-linear system applying the Theorem of Lagrange multipliers—to convert Discrete OP into Discrete OS—, for instance when direct search is used [78]. The real-time iteration scheme proposed in [3] is a numerical technique inspired by sequential quadratic programming (SQP) to solve (possibly) large-scale optimization problems arising from this approach, with real-time constraints. Differential dynamic programming (DDP) is within this approach when applied to a discrete-time dynamical systems [79]. DDP is based on Bellman's principle of optimality and then the optimal control is computed backward in time, starting from the final desired state. At each time step a linear or quadratic approximation of cost function and dynamics is considered, so that the optimal control is computed by solving a sequence of (possibly) constrained quadratic programming problems [7, 37].

Analytical strategy Impulse response of the liquid medium temperature field to a laser pulse can be computed analytically under certain assumptions, as described in §2.3.1. Thanks to linearity of heat equation and superposition principle, convolution integral yields to an (implicit) analytic solution of the thermal field given the laser beam trajectory. Given the sequence of past laser positions and emitted power, the actual thermal state of the liquid layer can be found. Thus, the system reduces to robot and droplet and the dynamical model involves ODEs only. It has to be underlined that the state of the system is not only the collection of robot and droplet states, because thermal dynamics is still present. The thermal state is hidden in the past laser history and there is no an explicit dynamics. Notice that at this point one may choose between optimization and time discretization. Actually, the latter choice defines what we refer to as *analytical strategy*; optimization at this step leads to involved

calculations[1] and has not been developed here. Time discretization corresponds to the approximation of laser continuous heating as a finite sum of laser pulses, as shown in §2.3.1. State and controls are parametrized also, and the OCP is transformed into a finite dimensional optimization problem (OP). It is worth noticing that there are less optimization variables w.r.t. the direct strategy (because thermal states are not explicit) but their relation is much more complicated (due to the hidden thermal dynamics). This approach is exploited in [74] employing both gradient-based and derivative-free optimizers to find a numerical solution.

During this work some of the aforementioned strategies have been developed and implemented. In particular, the *analytical strategy* is tested for off-line and on-line optimization, using gradient-based and direct search optimizers, in §4.3 and §4.4 respectively. Then the *direct strategy* based on FE discretization of the liquid medium is developed and implemented in §4.5 by using a direct search optimizer and simulation results are analysed.

4.3 OCP with analytical strategy

This Section faces the minimum time optimal control problem, Problem TOCP, formulated in §4.1 and follows the *analytical strategy* proposed in §4.2, based on the semi-analytical solution of the temperature field generated by the laser heating. The strategy is graphically resumed in Fig. 4.3.

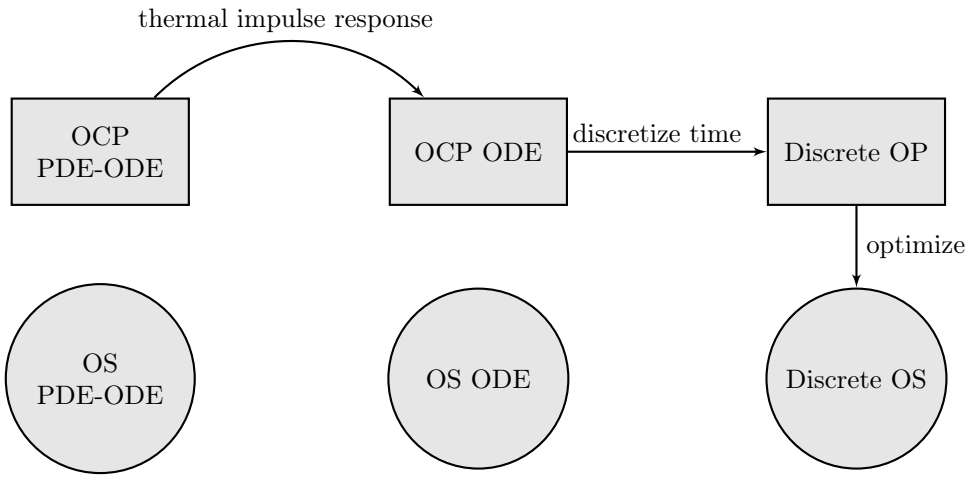


Figure 4.3: Flow chart of the *analytical strategy*.

Time transformation Problem TOCP is a free final time problem, because the final value T of the independent variable t is unknown *a priori*. Typically this type of problems is reformulated introducing the linear transformation $t = T\tau$, such that the

[1] Temperature field can be treated as a functional that takes laser trajectory and power as inputs. First order necessary optimality conditions are of difficult derivation; in particular the Du Bois lemma cannot be applied because the temperature field is given by a convolution integral.

problem is converted into a fixed final time problem if the normalized time $\tau \in [0,1]$ is considered as independent variable[2]. Cost functional, system dynamics and boundary conditions are re-written accordingly. Notice that parameter T is still an optimization variable and the cost functional remains $\mathcal{J} = T$; state and control constraints have to be satisfied $\forall \tau \in [0,1]$ and boundary conditions hold at $\tau = 0$ and $\tau = 1$. Dynamic constraint is expressed in terms of the new independent variable, namely $\forall \tau \in [0,1]$:

$$\mathbf{x}'(\tau) = T\,\boldsymbol{\varphi}(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau)) \tag{4.10}$$

The fixed final time optimal control problem, called τOCP, equivalent to Problem TOCP (4.3), is the following:

$$\begin{aligned} \text{minimize} \quad & T \\ \text{subject to} \quad & \mathbf{x}'(\tau) = T\,\boldsymbol{\varphi}(\tau, \mathbf{x}(\tau), \mathbf{u}(\tau)) \\ & \mathbf{x}(0) = \mathbf{x}^i \\ & \boldsymbol{B}_T \mathbf{x}(1) = \mathbf{x}_B^f \\ & \mathbf{x}_l \leq \mathbf{x}(\tau) \leq \mathbf{x}_u \\ & \mathbf{u}_l \leq \mathbf{u}(\tau) \leq \mathbf{u}_u \\ & T \geq 0 \\ \text{for} \quad & \tau \in (0,1) \end{aligned} \tag{4.11}$$

Dynamical model Let us recall the models of robot and droplet dynamics, respectively derived in §2.1 and §2.4:

$$\begin{aligned} \boldsymbol{M}_y \dot{\mathbf{y}}(t) &= \boldsymbol{A}_y\,\mathbf{y}(t) + \boldsymbol{B}_y\,\mathbf{u}^i(t) \\ \boldsymbol{M}_z \dot{\mathbf{z}}(t) &= \boldsymbol{A}_z(\mathbf{z}(t))\,\mathbf{z}(t) + \boldsymbol{B}_z\,\mathbf{g}_w(t, \mathbf{z}^x(t), \mathbf{y}, u^p) \end{aligned} \tag{4.12}$$

where $\mathbf{g}_w = \nabla_{\mathbf{x}} w$ is a functional that returns the gradient of temperature field $w = w(t, \mathbf{x})$. Control inputs $\mathbf{u} = [\mathbf{u}^i, u^p]^T \in \mathbb{R}^{n_u}$ are electric signals that drive the robot and laser power. Considering an equispaced time grid $\Gamma = \{t_k | t_k = k\Delta t, k = 0, 1, \ldots, N\}$ with time step $\Delta t = T/N$ and a sequence of laser pulses, the temperature gradient can be approximated to a finite sum, as explained in §2.3.1. Laser pulse at time t_j is described by robot position $\mathbf{y}_j^x$ and laser power u_j^p, $j = 0, 1, \ldots, N$; characteristic radius of the laser beam r_L is constant for $t \in [0, T]$. For every $t \in [0, T]$ there exists $J \in \{0, 1, \ldots, N\}$ such that $t_J \in \Gamma$ and $t_J \leq t < t_{J+1}$. Then, the temperature gradient in droplet position $\mathbf{z}^x$ is given by:

$$\mathbf{g}_w(t, \mathbf{z}^x) = -\frac{8}{\pi}\frac{\beta}{\varrho c_p} \sum_{j=0}^{J} \frac{u_j^p\,\Delta t}{[8\alpha(t - t_j) + r_L^2]^2} e^{-2\frac{\|\mathbf{z}^x - \mathbf{y}_j^x\|^2}{8\alpha(t-t_j)+r_L^2}} (\mathbf{z}^x - \mathbf{y}_j^x)^T \tag{4.13}$$

Hence, stacking robot and droplet states in vector $\boldsymbol{\chi} = [\mathbf{y}, \mathbf{z}]^T \in \mathbb{R}^{n_\chi}$, the dynamic model of the system is:

$$\boldsymbol{M}_\chi \dot{\boldsymbol{\chi}}(t) = \boldsymbol{A}_\chi(\boldsymbol{\chi}(t))\,\boldsymbol{\chi}(t) + \boldsymbol{B}_\chi\,\mathbf{u}(t) + \mathbf{b}_\chi(t, \boldsymbol{\chi}, \mathbf{u}) \tag{4.14}$$

for $t \in [0, T]$, where suitable matrices $\boldsymbol{M}_\chi$, $\boldsymbol{A}_\chi$ and $\boldsymbol{B}_\chi$ have to be introduced. Moreover, notice that $\mathbf{b}_\chi$ is a functional that takes as inputs the robot trajectory $\mathbf{y}^x$ and laser

[2] A notable exception is represented by the time optimal MPC proposed in [38], which minimizes the settling time of the system.

power u^p. The size of state $\boldsymbol{\chi}$ is $n_\chi = 8$ because it comprises position and velocity of both robot and droplet; the size of control $\mathbf{u}$ is $n_u = 3$ or $n_u = 2$ depending on whether laser power is controlled or not. Initial and final conditions can be written as in §4.1, with matrix $\boldsymbol{B}_T$ selecting constraint states at the final time:

$$\boldsymbol{\chi}(0) = \boldsymbol{\chi}^i \ , \qquad \boldsymbol{B}_T \boldsymbol{\chi}(T) = \boldsymbol{\chi}_B^f \tag{4.15}$$

By applying the linear time transformation $t = T\tau$ and inverting (diagonal) matrix $\boldsymbol{M}_\chi$ the dynamics can be expressed as in §4.1, introducing the functional $\boldsymbol{\psi}_\tau$:

$$\boldsymbol{\chi}'(\tau) = \boldsymbol{\psi}_\tau \left(\tau, T, \boldsymbol{\chi}, \mathbf{u}\right) \tag{4.16}$$

whose explicit expression is:

$$\boldsymbol{\psi}_\tau(\tau, T, \boldsymbol{\chi}, \mathbf{u}) = T \begin{bmatrix} \mathbf{z}^v \\ \dfrac{b_D}{m_D}\mathbf{g}_w(T\tau, \mathbf{z}^x, \mathbf{y}^x, u^p) - \dfrac{c_D}{m_D}\|\mathbf{z}^v\|\mathbf{z}^v \\ \mathbf{y}^v \\ \dfrac{b_L}{m_L}\mathbf{u}^i - \dfrac{c_L}{m_L}\mathbf{y}^v \end{bmatrix} \tag{4.17}$$

Similarly for boundary conditions at $\tau = 0$ and $\tau = 1$, it is:

$$\boldsymbol{\chi}(0) = \boldsymbol{\chi}^i \ , \qquad \boldsymbol{B}_T \boldsymbol{\chi}(1) = \boldsymbol{\chi}_B^f \tag{4.18}$$

As noticed before, system dynamics depends on the past history because temperature field has to be computed considering all the past laser pulses. Actually, effect of past pulses becomes less important as distance in time increases. Thus, in order to lighten the computational effort, it may be advisable to consider only laser pulses within a fixed-length time interval in the past, ensuring that previous pulses have a negligible effect on the actual dynamics.

Time discretization Let us build a time grid Γ_τ over the normalized time domain $\tau \in [0, 1]$, equispaced for simplicity with step size $\Delta\tau = 1/N$.

$$\Gamma_\tau = \{\tau_k | \tau_k = k\Delta\tau \, , k = 0, 1, \ldots, N\} \tag{4.19}$$

The corresponding (physical) time grid Γ depends on the *a priori* unknown final time T, and the time step size is $\Delta t = T\Delta\tau = T/N$. Parameter N has to be carefully chosen before solving the problem, when a value for the final time T is not available. Also, it is advisable to select a suitable value for N because it affects the accuracy of the approximation of system dynamics. On the other hand, large values of N lead to high computational effort and time. Thus, *a priori* knowledge should be employed to have a good estimate of T and then a suitable choice of N.

Let us adopt a piece-wise constant parametrization of controls and forward finite difference scheme for state discretization. Denoting the approximation $\boldsymbol{\chi}(\tau_k) \approx \boldsymbol{\chi}_k$, the update equation is:

$$\boldsymbol{\chi}_{k+1} = \boldsymbol{\chi}_k + \Delta\tau \, \boldsymbol{\psi}_\tau(\tau_k, T, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_{0,\ldots,k}) \tag{4.20}$$

where $\boldsymbol{\chi}_{0,\ldots,k} = \{\boldsymbol{\chi}_j | j = 0, \ldots, k\}$ and $\mathbf{u}_{0,\ldots,k} = \{\mathbf{u}_j | j = 0, \ldots, k\}$ collect states and controls.

Direct transcription Standard optimizers, e. g. IPOPT and `fmincon` provided by Matlab, are designed to find (local) solutions of NLP problems of the form:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) \geq \mathbf{0} \\ & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \end{aligned} \tag{4.21}$$

where f, $\mathbf{g}$ and $\mathbf{h}$ are objective, equality constraints and inequality constraints functions respectively. Vectors $\mathbf{x}_l$ and $\mathbf{x}_u$ denote lower and upper bounds on variables in $\mathbf{x}$; inequalities hold component wise.

Let us define the vector of optimization variables $\mathbf{x} \in \mathbb{R}^{n_x}$ composed by final time, states and controls at each time step, whose size is $n_x = 1 + (N+1)(n_\chi + n_u)$. In the case $N = 100$ and $n_u = 2$, the NLP has $n_x = 1011$ optimization variables.

$$\mathbf{x} = [T, \boldsymbol{\chi}_0, \ldots, \boldsymbol{\chi}_N, \mathbf{u}_0, \ldots, \mathbf{u}_N]^T \tag{4.22}$$

Cost functional of Problem TOCP is $\mathcal{J} = T$, hence objective function is $f(\mathbf{x}) = T$. Equality constraints $\mathbf{g}(\mathbf{x})$ impose initial[3]and final conditions and make the system to evolve satisfying the discrete-time dynamic model.

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\chi}_0 - \boldsymbol{\chi}^i \\ \boldsymbol{B}_T \boldsymbol{\chi}_N - \boldsymbol{\chi}_B^f \\ \boldsymbol{\chi}_1 - \boldsymbol{\chi}_0 - \Delta\tau \boldsymbol{\psi}_\tau(\tau_0, T, \boldsymbol{\chi}_0, \mathbf{u}_0) \\ \boldsymbol{\chi}_2 - \boldsymbol{\chi}_1 - \Delta\tau \boldsymbol{\psi}_\tau(\tau_1, T, \boldsymbol{\chi}_{0,1}, \mathbf{u}_{0,1}) \\ \vdots \\ \boldsymbol{\chi}_N - \boldsymbol{\chi}_{N-1} - \Delta\tau \boldsymbol{\psi}_\tau(\tau_{N-1}, T, \boldsymbol{\chi}_{0,\ldots,N-1}, \mathbf{u}_{0,\ldots,N-1}) \end{bmatrix} \tag{4.23}$$

Bounds on droplet and robot states and controls, e. g. maximum robot velocities or motor currents, are embedded into the problem using simple boxes in the form $\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u$ instead of $\mathbf{h}(\mathbf{x}) \geq \mathbf{0}$. This may become infeasible with complex geometries of the liquid domain or strong coupling between states and/or controls. Equality constraints comprise boundary conditions and discrete-time dynamics, thus $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^{n_g}$ with $n_g = (N+2)n_\chi$. In the case $N = 100$, the NLP has $n_g = 816$ equality constraints. Inequality constraints may be adopted to enforce final conditions in a softer way w.r.t. equality constraints, for instance to move the droplet in a given area around a point, instead of exactly in a given point. The same idea could be exploited by using simple bounds, e. g. to move the droplet in a rectangular domain.

Gradient and Jacobian Gradient-based optimizers may take advantage of explicit expressions for the first and second derivatives of objective and constraint functions f

[3]Equality constraint for initial conditions may be explicitly solved for the initial state, $\boldsymbol{\chi}_0 = \boldsymbol{\chi}^i$, and so initial condition $\boldsymbol{\chi}^i$ can be used instead of the initial state $\boldsymbol{\chi}_0$. Actually, this would eliminate a trivial constraint and lighten a bit the computation. On the other hand, as proposed in [3], in MPC applications it may be useful and more powerful to consider the initial condition as a parameter of the problem and the initial state as an optimization variable. In [3] this approach is called *initial value embedding*. The key feature is that derivatives w.r.t. the initial state are available. Between consecutive OCPs in MPC iterations, initial conditions may slightly change, following the system evolution. Convergence of the warm-started algorithm to approximate the solution of each OCPs is typically faster whenever derivatives can be easily computed.

and $\mathbf{g}$ with respect to optimization variables $\mathbf{x}$. First and second derivatives of f are easily obtained for the pure minimum time problem:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \tag{4.24}$$

$$\nabla^2_{\mathbf{x}} f(\mathbf{x}) = \mathbf{0} \tag{4.25}$$

Jacobian matrix of the constraint function $\mathbf{g}$ w.r.t. optimization variables $\mathbf{x}$ can be written in explicit form, namely:

$$\nabla_{\mathbf{x}} \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial \mathbf{g}}{\partial T} & \nabla_{\boldsymbol{\chi}_0} \mathbf{g} & \dots & \nabla_{\boldsymbol{\chi}_N} \mathbf{g} & \nabla_{\mathbf{u}_0} \mathbf{g} & \dots & \nabla_{\mathbf{u}_N} \mathbf{g} \end{bmatrix} \tag{4.26}$$

Its derivation is reported for completeness in Appendix B while the Hessian of $\mathbf{g}$ has not been calculated explicitly. Notice that the Jacobian matrix $\nabla_{\mathbf{x}} \mathbf{g}(\mathbf{x})$ has $n_g n_x$ entries, thus its size grows quadratically with respect to the number of nodes in the time grid.

$$n_g n_x = (N+2) n_\chi \left[1 + (N+1)(n_\chi + n_u)\right] = \mathcal{O}(N^2) \tag{4.27}$$

4.3.1 Numerical solution

Mathematical optimization problems, like the one in (4.21), have been widely investigated and many different techniques are available to find a solution, perhaps numerical and approximate. One may use algorithms that takes a finite number of steps to find a solution, or iterative methods that converge to a solution or other heuristics.

Non-linear programming problems often require iterative or heuristic methods. Heuristics is a class of algorithms which cannot guarantee to find a solution; typically these algorithms gain a trade-off between optimality, computational time and completeness and may be useful from the practical point-of-view. Iterative methods generate successive improved approximations of the solution starting from an initial guess. These methods may use informations about the problem, i. e. they evaluate Hessians, gradient or only function values. Evaluating derivatives may improve the rate of convergence but also increases the computational cost of each iteration. Newton's method, sequential quadratic programming (SQP) and interior point methods require to evaluate Hessians, or to approximate Hessians, e. g. using finite differences. Quasi-Newton's, steepest descent, conjugate gradient and coordinate descent methods use gradients or approximated gradients only. Finally, methods that use only function values are called direct search methods; for instance these comprise pattern search and Nelder-Mead method (also known as simplex method).

The best method and algorithm depend on the particular problem and the choice should be influenced by the relative computational cost for evaluating objective and constraint functions and their derivatives.

Gradient-based optimizer Analytical expressions of objective function gradient and constraint function Jacobian have been derived explicitly. Most of the computational effort is in calculating thermal gradient and its derivatives. It is highlighted from Equations (B.16)-(B.17)-(B.19)-(B.20) in Appendix B that the computational costs to evaluate temperature gradient and its Jacobian matrix are comparable — the Jacobian matrix is sparse but likely requires more memory. Thus it is probably more efficient to use a gradient-based optimizer and to evaluate the analytical expression of the Jacobian instead of using a finite difference approximation (that uses more than one function evaluation).

Table 4.1: Reference parameters of robot model.

m_L	1	kg
c_L	2	N s/m
b_L	10	N/A
$i^{\max}$	1	A
$v_L^{\max}$	0.5	m/s

Initial guess Iterative optimization algorithms typically requires an initial guess, that is an estimate of the solution, to start the iterating process. As proposed and implemented in [74], an initial guess can be generated solving a reference tracking problem, like Problem TMPC proposed in §4.1, that requires none or a much simpler initial guess. A feasible approach to this problem is reported and discussed in §4.4.

If trajectory optimization is not embedded into the problem, a reference path has to be generated. In simple domains, e. g. when the shape of the liquid layer is convex, and no obstacles are present, the reference path may be the line connecting the actual position to the target position of the droplet. With more complex scenarios the definition of a reference path may be quite involved. A possible approach to face this problem is to simulate the evolution of system with a *cold* laser beam that refrigerate the target position — extracting heat instead of releasing heat. The induced temperature gradient attracts and eventually moves the droplet toward the desired position, overtaking obstacles and corners. It is interesting to notice that using this ideal *cold* laser beam the control mechanism is stable and much robust (unfortunately its practical implementation is difficult at the moment).

Finally, it should be noticed that solution to Problem TMPC could be found neglecting the robot dynamics and using laser position as control input. Then original robot controls and states can be reconstructed by means of the robot inverse dynamics. This procedure is feasible because the robot model is simple and treatable enough.

4.3.2 Simulation results

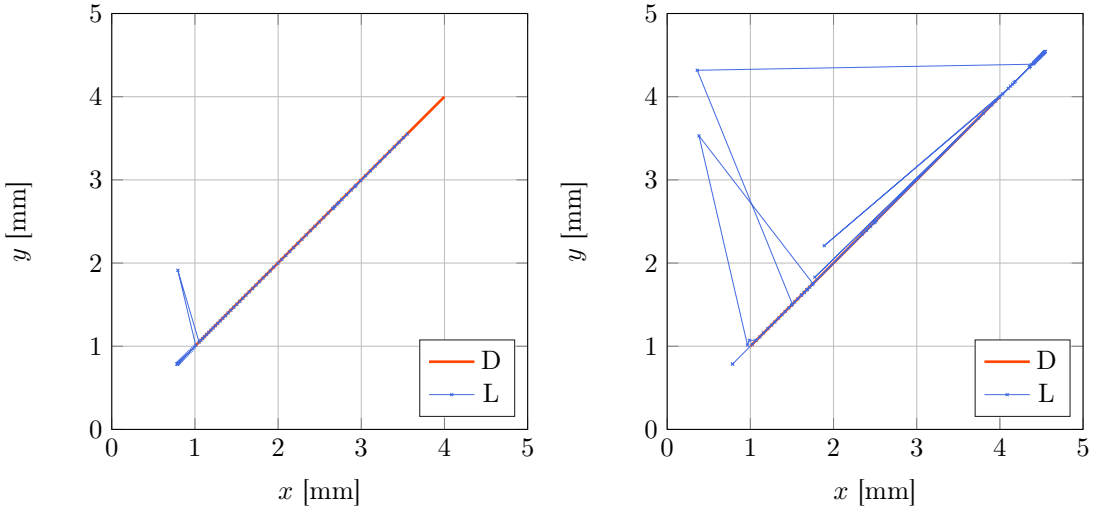
The `fmincon` routine provided by Matlab has been used to solve (4.21), Tab. 4.2. We highlight the fact that reported results may correspond to local optima found as solution of the OCP. Liquid and droplet properties and reference robot parameters are reported in Tab. 3.2 and Tab. 4.1 respectively. Time step size $\Delta t = T/N$ and the number of optimization variables $n_x = 1 + 10(N + 1)$ depend on the number of nodes N and the final time T. The latter is known *a posteriori*, while the number of nodes only after the generation of a guess, that uses a fixed time step and stops when the final point is reached within a given tolerance.

Tests The approach has been tested with two particular final conditions, viz. free and zero final velocity of the droplet; final state of the robot is left free. Initially both droplet and robot have null velocity and the latter is in $\mathbf{y}^x = [0, 0]^T$. Initial and final conditions have been enforced with the equality constraint functions—hence they are hard constraints.

Guess Problem TMPC (4.7) has been solved to generate an initial guess, using the `patternsearch` routine provided by Matlab, as explained in §4.4. Reference path is a

Table 4.2: Options for `fmincon` in Matlab.

Algorithm	*interior-point*
HessianApproximation	*BFGS*
MaxIterations	300
MaxFunctionEvaluations	1500
ConstraintTolerance	1×10^{-12}
OptimalityTolerance	1×10^{-3}



(a) *Free final velocity.* **(b)** *Zero final velocity.*

Figure 4.4: Droplet and laser paths.

straight line from the initial to the final position. The time step used for generating the guess solution is $\Delta t = 50\,\text{ms}$, but this only defines the refinement of the normalized time grid, and the time horizon $T_{\text{horiz}} = 1\,\text{s}$. Optimizer option on mesh tolerance has been set $TolMesh = 10\,\mu\text{m}$, being a trade-off between accuracy and speed in guess solution generation.

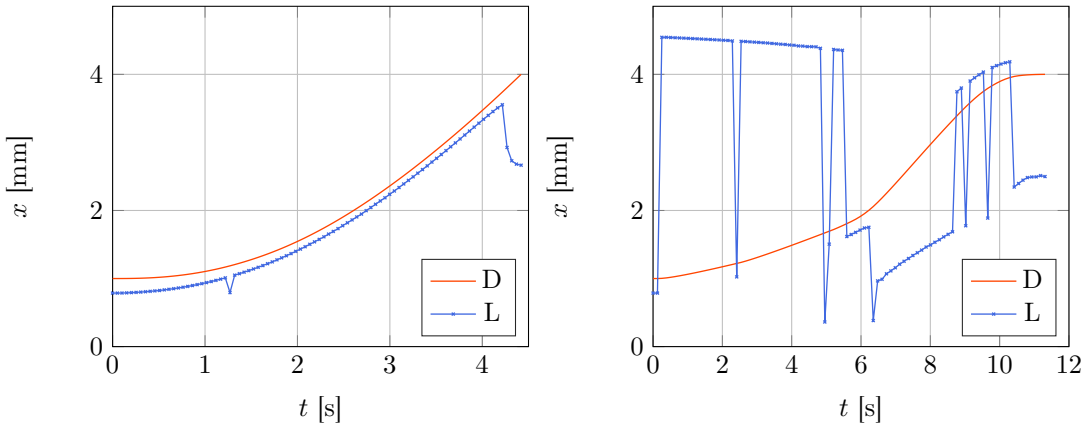
Results and discussion The solution found by the optimizer may slightly violate dynamical constraints. Thus, optimal controls are selected from the vector of optimized variables and are used to integrate the system dynamics to obtain an evolution that strictly satisfies constraints [74].

Paths of droplet and laser for free and zero final droplet velocity are reported graphically in Fig. 4.4. Then, in Fig. **??** and Fig. 4.6a positions and velocities along the x axis are shown.

Let us discuss about the case of free final velocity, where the laser always pushes the droplet. Looking at Fig. 4.6b it is clear that there exists a droplet-laser distance that maximizes the velocity of migration. One may analyse this problem from the point-of-view of §3.2.3. However, steady state conditions are not reached in the simulated time

Table 4.3: Results for different final conditions on droplet velocity.

	free	zero	
N	87	113	
n_x	881	1141	
T	4.50	12.33	s
Δt	0.041	0.109	s

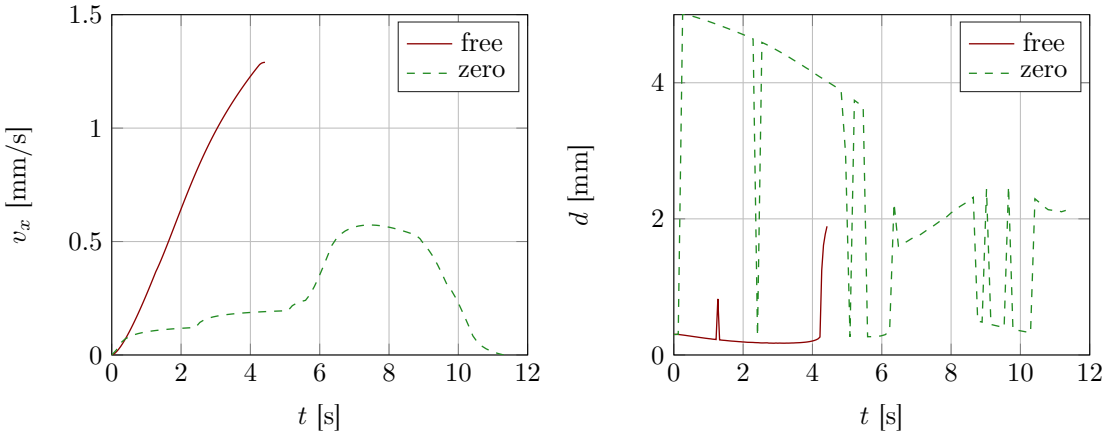


(a) *Free final velocity.* **(b)** *Zero final velocity.*

Figure 4.5: Droplet and laser position along the x axis.

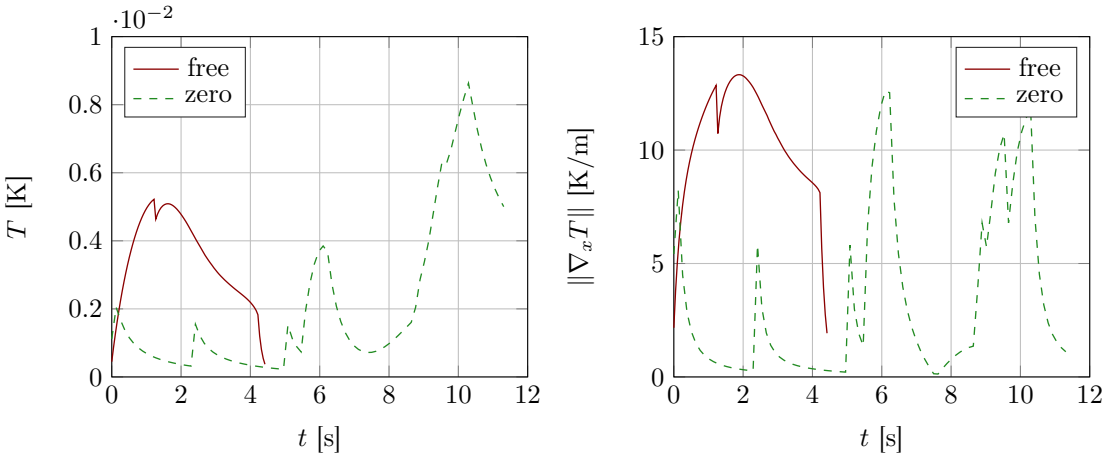
slot; actually, it seems that the droplet is uniformly accelerated, because of the almost linear speed profile, Fig. 4.6a. Droplet temperature initially arises due to the laser heating, Fig. 4.7a, but then decreases as droplet (and robot) speed increases—because less energy per unit volume is released at higher speed. Simultaneously, temperature gradient exhibits almost the same behaviour, Fig. 4.7b. Then, after stopping the laser heating, droplet is subject to drag and thermocapillary force induced by the residual unsteady temperature field, so without the dominant driving force the droplet slows down [74].

Considering the case of zero final velocity, the optimal strategy is quite different. As reported in Fig. 4.4 for zero final velocity, the robot moves repeatedly from behind the droplet to the neighbourhood of the target position. In particular, the robot starts pushing the droplet, it moves close to the target—after the target w.r.t. the droplet—, then it pushes again the droplet and finally it adjusts the trajectory, Fig. 4.5. So it resembles that the optimal strategy takes advantage of the slow thermal dynamics to prepare in advance the thermal gradient needed to stop the droplet. We expect that the timing of these steps would depend on the time constants of the system, in particular the ratio between droplet and temperature time constants. It is notable that in the pushing phase (around $t \approx 6\,\mathrm{s}$ and $t \approx 10\,\mathrm{s}$ in Fig. 4.6b) the laser-droplet distance is quite similar to the quasi-constant distance in the case of free final velocity. Temperature profile in this case has a maximum close to the final time, because of the initial heating to brake the droplet at the end, and it is higher than the peak in the case of free final velocity, Fig. 4.7a. Conversely, temperature gradient is much



(a) *Droplet velocity.* **(b)** *Droplet-laser distance.*

Figure 4.6: Droplet velocity along the x axis and droplet-laser distance for free and zero final velocity.



(a) *Temperature.* **(b)** *Temperature gradient.*

Figure 4.7: Temperature and temperature gradient in droplet position for free and zero final velocity.

smaller on average—otherwise higher speed is reached and higher temperature gradient is needed to brake—, Fig. 4.7b. Finally, we can notice the rupture of symmetry in the laser spot trajectory, Fig. 4.4. Only few points are captured (due to large robot acceleration and large time step) but it is clear that the laser would shut down in those points, in order to not affect droplet dynamics [74].

4.4 MPC with analytical strategy

This Section faces the reference tracking optimal control problem, Problem TMPC, formulated in §4.1 and follows the *analytical strategy* proposed in §4.2 and exploited in §4.3. The strategy is graphically resumed in Fig. 4.3.

The denomination of Problem TMPC refers to the fact that this problem may be solved on-line to generate a model-based feedback controller, that is the basic idea of MPC.

Dynamical model Let us recall the mathematical model of robot and droplet dynamics, Eq. (4.12), expressed in Eq. (4.14) in terms of vector $\boldsymbol{\chi} = [\mathbf{y}, \mathbf{z}]^T$. The dynamic model for $t \in [0, T]$ reads:

$$\dot{\boldsymbol{\chi}}(t) = \boldsymbol{\psi}(t, \boldsymbol{\chi}, \mathbf{u}) \tag{4.28}$$

where functional $\boldsymbol{\psi}$ accounts for the subsumed thermal field.

Curvilinear coordinates Let us focus on the kinematics of curvilinear coordinates, that has to be integrated to evaluate the cost functional $\mathcal{J}_{\text{MPC}}$ of Problem TMPC (4.7). A given reference path γ, i. e. a planar smooth curve, is parametrized by the curvilinear abscissa $s \in [0, L]$, where $L \in (0, +\infty)$ is the curve length. The curve γ is described by its curvature $\varrho_\gamma(s)$, $s \in [0, L]$; initial point $[x_\gamma(0), y_\gamma(0)]$ and orientation $\vartheta_\gamma(0)$ permit to locate every point of γ in a fixed absolute reference frame. In particular, the given path can be expressed in a Cartesian coordinate system by integrating:

$$\begin{aligned} x_\gamma(s) &= x_\gamma(0) + \int_0^s \cos\vartheta_\gamma(s)\,\mathrm{d}s \\ y_\gamma(s) &= y_\gamma(0) + \int_0^s \sin\vartheta_\gamma(s)\,\mathrm{d}s \\ \vartheta_\gamma(s) &= \vartheta_\gamma(0) + \int_0^s \varrho_\gamma(s)\,\mathrm{d}s \end{aligned} \tag{4.29}$$

Droplet position can be represented using its distance n from the nearest point on the curve and the curvilinear abscissa s of that point. There is no meaning in defining the orientation of a point-mass droplet, hence vector $\boldsymbol{\xi} = [s, n]^T$ completely describes the droplet configuration. Let us assume that for any given droplet position $\mathbf{z}^x = [x_D, y_D]^T$ and curve γ there exist $\boldsymbol{\xi} = [s, n]^T$ such that the following hold:

$$\begin{aligned} x_D &= x_\gamma(s) - n\sin\vartheta_\gamma(s) \\ y_D &= y_\gamma(s) + n\cos\vartheta_\gamma(s) \end{aligned} \tag{4.30}$$

Assumption (4.30) is verified if curve γ is smooth enough. Actually, it may happen that more than one point $[x_\gamma(s), y_\gamma(s)]$ on the curve satisfies the condition (4.30).

Let us analyse the evolution of droplet position in curvilinear coordinates. Considering time-varying droplet position, $\mathbf{z}^x(t) = [x_D(t), y_D(t)]^T$, the first time derivative leads to (omitting time dependence):

$$\begin{aligned} \dot{x}_D &= \left(\frac{\mathrm{d}x_\gamma}{\mathrm{d}s}(s) - n\cos\vartheta_\gamma(s)\frac{\mathrm{d}\vartheta_\gamma}{\mathrm{d}s}(s) \right) \dot{s} - \sin\vartheta_\gamma(s)\dot{n} \\ \dot{y}_D &= \left(\frac{\mathrm{d}y_\gamma}{\mathrm{d}s}(s) - n\sin\vartheta_\gamma(s)\frac{\mathrm{d}\vartheta_\gamma}{\mathrm{d}s}(s) \right) \dot{s} + \cos\vartheta_\gamma(s)\dot{n} \end{aligned} \tag{4.31}$$

Thus, the relationship between velocities in Cartesian and curvilinear coordinates is linear; substituting (4.29) it reduces to:

$$\begin{bmatrix} \dot{x}_D \\ \dot{y}_D \end{bmatrix} = \begin{bmatrix} \cos\vartheta_\gamma(s)\,(1 - n\varrho_\gamma(s)) & -\sin\vartheta_\gamma(s) \\ \sin\vartheta_\gamma(s)\,(1 - n\varrho_\gamma(s)) & \cos\vartheta_\gamma(s) \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{n} \end{bmatrix} \tag{4.32}$$

and, whenever $n \neq 1/\varrho_\gamma(s)$, the inverse transformation is obtained by inverting the matrix:

$$\begin{bmatrix} \dot{s} \\ \dot{n} \end{bmatrix} = \begin{bmatrix} \dfrac{\cos\vartheta_\gamma(s)}{1 - n\varrho_\gamma(s)} & \dfrac{\sin\vartheta_\gamma(s)}{1 - n\varrho_\gamma(s)} \\ -\sin\vartheta_\gamma(s) & \cos\vartheta_\gamma(s) \end{bmatrix} \begin{bmatrix} \dot{x}_D \\ \dot{y}_D \end{bmatrix} \tag{4.33}$$

Finally, kinematics of curvilinear coordinates can be written as:

$$\dot{\boldsymbol{\xi}}(t) = \boldsymbol{R}_\gamma(\boldsymbol{\xi}(t))\,\mathbf{z}(t) \quad t \in [0, T] \tag{4.34}$$

and it can be integrated in parallel to droplet motion, in order to update its relative position w.r.t. the desired path γ and to evaluate the cost functional $\mathcal{J}_{\text{MPC}}$ (4.6).

Time discretization Let us build a time grid Γ over the time domain $t \in [0, T]$, equispaced for simplicity with step size $\Delta t = T/N$. Final time T is typically called time horizon in the MPC framework.

$$\Gamma = \{t_k | t_k = k\Delta t\,, k = 0, 1, \ldots, N\} \tag{4.35}$$

Time step size Δt, and so number of points N for a given time horizon, affects the accuracy of the discrete-time evolution and the computational time for the integration. Control parametrization and state discretization have been already discussed for system dynamics in §4.3. Denoting the approximation $\boldsymbol{\chi}(t_k) \approx \boldsymbol{\chi}_k$, the update equation of system state is:

$$\boldsymbol{\chi}_{k+1} = \boldsymbol{\chi}_k + \Delta t\,\boldsymbol{\psi}(t_k, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_{0,\ldots,k}) \tag{4.36}$$

Let us consider the discrete-time update of curvilinear coordinates $\boldsymbol{\xi}(t_k) \approx \boldsymbol{\xi}_k$ for $k \in \{0, 1, \ldots, N\}$. Adopting Heun's method an intermediate estimate $\hat{\boldsymbol{\xi}}_{k+1}$ of the next state is calculated by using the explicit Euler's scheme, Eq. (4.37). Then, trapezoidal rule refines the final estimate $\boldsymbol{\xi}_{k+1}$ at the next integration point, Eq. (4.38).

$$\hat{\boldsymbol{\xi}}_{k+1} = \boldsymbol{\xi}_k + \Delta t\,\boldsymbol{R}_\gamma(\boldsymbol{\xi}_k)\,\mathbf{z}_k \tag{4.37}$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \frac{\Delta t}{2}\left[\boldsymbol{R}_\gamma(\hat{\boldsymbol{\xi}}_{k+1})\,\mathbf{z}_{k+1} + \boldsymbol{R}_\gamma(\boldsymbol{\xi}_k)\,\mathbf{z}_k\right] \tag{4.38}$$

Update rule (4.37)–(4.38) can be applied only if the next droplet state $\mathbf{z}_{k+1}$ is available. Thus, system state has to be updated with Eq. (4.36) and then curvilinear coordinates can be updated accordingly.

Notice that Heun's method has been chosen because it leads to an explicit update equation. Trapezoidal rule arises if the intermediate estimate $\hat{\boldsymbol{\xi}}_{k+1}$ in (4.38) is replaced by the actual value of $\boldsymbol{\xi}_{k+1}$. In this case an implicit equation has to be solved to find $\boldsymbol{\xi}_{k+1}$. Reference paths for droplet manipulation are reasonably supposed to be smooth, then Heun's method and trapezoidal rule are expected to return similar estimations. Thus, the explicit method is favoured in this case because it requires less computational effort and gives similar results.

Cost functional Let us focus on cost functional $\mathcal{J}_{\text{MPC}}$, defined in Eq. (4.6), that takes into account both speed and accuracy of tracking. In a discrete-time settings, it has to be approximated by using a time grid and a quadrature rule. For the sake of simplicity, let us consider the time grid Γ defined above (so that interpolation is not needed) with uniform time step Δt and adopt the trapezoidal rule, neglecting the error term of order $\mathcal{O}(\Delta t^2)$.

$$\mathcal{J}_{\text{MPC}} = -s(T) + \omega_n \int_0^T n^2(t)\,\mathrm{d}t \approx -s_N + \omega_n \frac{\Delta t}{2} \sum_{k=0}^{N-1} \left(n_k^2 + n_{k+1}^2\right) \tag{4.39}$$

where $s_k \approx s(t_k)$ and $n_k \approx n(t_k)$ for $t_k \in \Gamma$ are computed with the discrete-time dynamics (4.36)-(4.37)-(4.38).

Control input In practice, it is interesting to analyse the case when robot dynamics is much faster than droplet dynamics. In this scenario, one could isolate the robot dynamics in order to simplify the OCP. The optimal control of this simplified OCP is a trajectory of the laser beam that moves the droplet as desired.

Actually, this trajectory can be a reference for a low-level controller that drives the robot. Another approach for robot control is to employ its inverse dynamics and to reconstruct the control inputs to generate the desired trajectory (if feasible). This strategy can be exploited here because model in §2.1 is simple enough.

Given a sequence of laser positions $\{\mathbf{x}_{L,k} | k = 0, 1, \ldots, N\}$ with $\mathbf{x}_{L,k} \approx \mathbf{x}_L(t_k)$, velocity $\mathbf{v}_{L,k}$ and acceleration $\mathbf{a}_{L,k}$ can be estimated. Omitting the errors due to the forward finite difference approximation, for $k \in \{1, \ldots, N-1\}$ they are given by:

$$\begin{aligned} \mathbf{v}_{L,k} &= \frac{\mathbf{x}_{L,k+1} - \mathbf{x}_{L,k}}{\Delta t} \\ \mathbf{a}_{L,k} &= \frac{\mathbf{x}_{L,k+1} - 2\,\mathbf{x}_{L,k} + \mathbf{x}_{L,k-1}}{\Delta t^2} \end{aligned} \tag{4.40}$$

Values for $k \in \{0, N\}$ can be found using other approximations, e. g. backward finite difference scheme. Substituting $\dot{\mathbf{y}}^v(t_k) = \mathbf{a}_{L,k}$ and $\mathbf{y}^v(t_k) = \mathbf{v}_{L,k}$ in robot model (2.5), an estimate of robot control input $\mathbf{u}_k^i \approx \mathbf{u}^i(t_k)$ can be found, namely:

$$\mathbf{u}_k^i = \frac{c_L}{b_L}\mathbf{v}_{L,k} + \frac{m_L}{b_L}\mathbf{a}_{L,k} \tag{4.41}$$

The sequence of control inputs $\{\mathbf{u}_k^i | k = 0, 1, \ldots, N\}$ may be used as a guess to initialize the numerical solution of Problem TOCP with robot dynamics, §4.3.

4.4.1 Numerical solution

In MPC both accuracy and speed of the solution are considered, because of optimality of the control loop and time constraints. In this setting typically a fast, perhaps

Table 4.4: MPC parameters.

T_{horiz}	2	s
Δt	0.04	s
T_{MPC}	0.04	s
Δt_{MPC}	0.04	s
ω_n	3.33×10^4	$(\text{m s})^{-1}$

approximate, response is needed; after one time step another solution will be computed in order to compensate for disturbances and inaccuracies in the model [3, 5, 39, 80]. To this end, it is advisable to simplify as much as possible the OCP.

Here we neglect the robot dynamics, as suggested above, thus optimization variables are laser positions. Moreover, we strongly limit the optimization space introducing a constraint on the control sequence: future laser pulses have to be in the same position $\mathbf{x}_L^*$ and laser switches off after N_L^* pulses, where N_L^* is specified by the user. With these limitations, optimization variables are in a 2-dimensional space. Initial guess for the next optimal control $\mathbf{x}_L^*$ could be the actual droplet position $\mathbf{z}^x$ or laser position $\mathbf{y}^x$, or a position in between. Then, additional bounds could be imposed to take into account the robot dynamics (2.5). For instance, robot velocity can be estimated by using the finite difference approximation and then the displacement of the next time step can be limited accordingly.

With these assumptions, Problem TMPC has only two optimization variables, constrained with simple bounds. It may be convenient to adopt a derivative-free optimizer to take advantage of the small optimization space, that is sampled to evaluate the system evolution and then the corresponding cost. Optimization for nonlinear MPC through a direct search approach has been recently discussed in [78]. The `patternsearch` routine provided by Matlab handles linear and non-linear equality and inequality constraints and it has been used to generate results reported in this Section.

The MPC runs every T_{MPC} and evaluates the cost functional over a time horizon T_{horiz}, integrating the discrete-time dynamics of the system with time step Δt_{MPC}. Given the solution, the corresponding control input is fed into the system, whose evolution is computed with a (typically smaller) time step Δt. Discrete-time update rules are obtained by the trapezoidal rule, as explained in §2.3.2 and §3.4. Robot dynamics is not considered in system evolution, thus laser positioning is instantaneous.

4.4.2 Simulation results

Tests The approach has been tested with the reference path shown in Fig. 4.8 and free final condition, i. e. considering cost functional $\mathcal{J}_{\text{MPC}}$ (4.6). Initial droplet position is given, while robot is placed initially according to the solution of the first MPC run. The initial guess of each optimization process is the actual droplet position. Moreover, the optimal laser position is constrained to be in a rectangular domain around the reference path.

Results and discussion Let us discuss about the resulting evolution of the system, with the MPC loop and instantaneous robot dynamics. Looking at Fig. 4.8 the control strategy seems confused in the first part, when the droplet is close to the corner, and

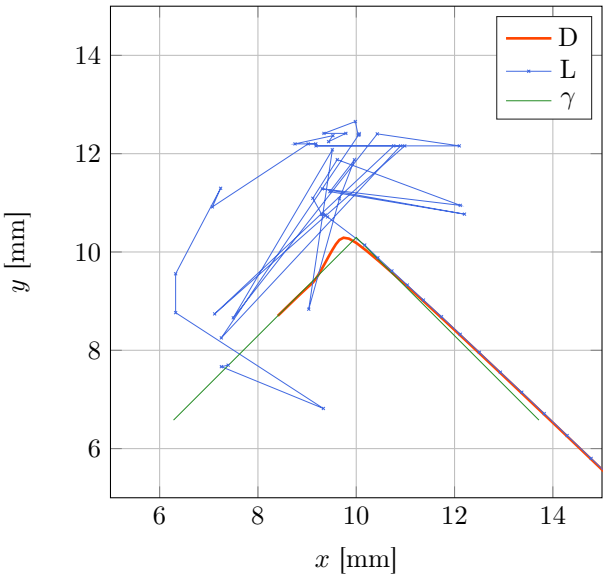
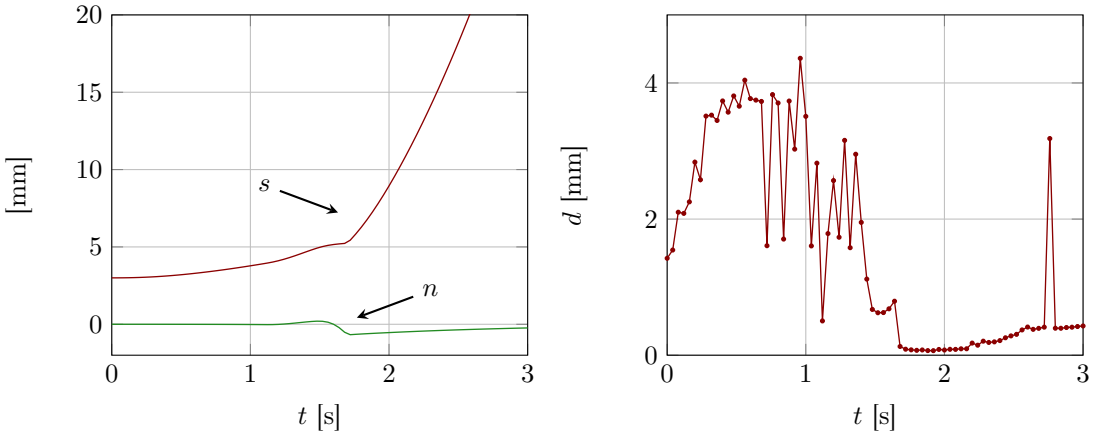


Figure 4.8: Reference tracking with no final constraints: droplet and laser path.

then more intuitive during the final part, when the laser pushes the droplet. Initially the droplet is pushed forward and then laser goes outside the corner to induce a temperature gradient able to move the droplet along the reference path. Then, the same strategy of maximum pushing found in §4.3 in the case of free final velocity is exploited, Fig. 4.10a. Droplet-laser distance in the final part is comparable with values found above, Fig. 4.9b. Tracking performance in terms of maximum lateral deviation is good, $\max |n| < 1\,\text{mm}$, as evidenced in Fig. 4.9a. However, smaller deviations can be achieved setting higher values of ω_n in the cost functional $\mathcal{J}_{\text{MPC}}$ to penalise the lateral error.

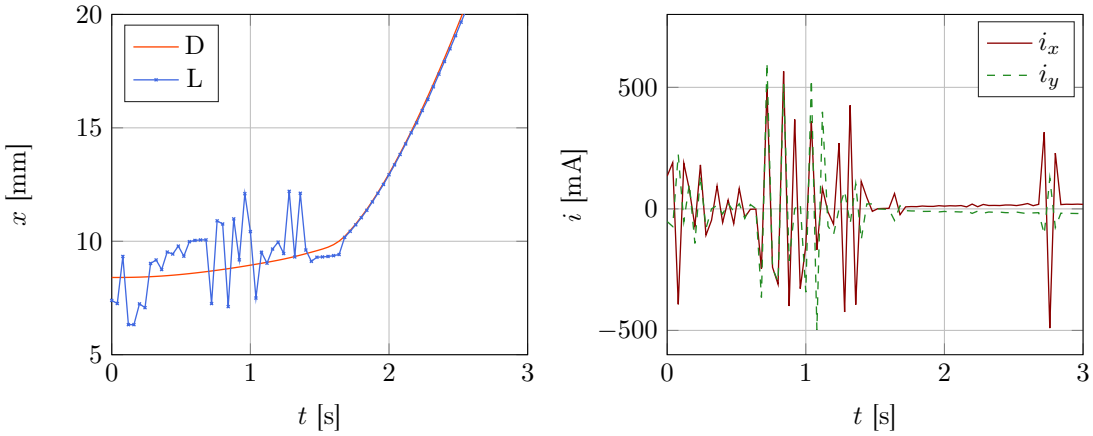
Robot movements are instantaneous, in the sense that laser position is exactly the solution of the previous MPC problem, with no filtering due to robot dynamics. However, it is possible to reconstruct equivalent control inputs that would generate the laser trajectory shown in Fig. 4.8. Finite difference approximation of laser velocity and acceleration, reference robot parameters in Tab. 4.1, and Eq. (4.41) lead to the equivalent controls, reported graphically in Fig. 4.10b.

Finally, a last consideration about the (coupled) effect of time horizon and control parametrization. Let us look at Fig. 4.8 and Fig. 4.9a. Before the corner the laser does not push at maximum and the droplet moves slowly. This happens because the controller takes into account few laser pulses and then free evolution of the system (this is the control parametrization). In this situation the time horizon is too long, in the sense that the free evolution moves the droplet away from the path. This inhibits the laser to push and slow down the motion. Then, close to the corner, the laser induces a temperature gradient outside the corner to change droplet velocity direction. In the end, the laser pushes as much as possible the droplet along the straight line—the optimal laser-droplet distance is almost reached, Fig. 4.9b.



(a) *Curvilinear coordinates.* **(b)** *Droplet-laser distance.*

Figure 4.9: Reference tracking with no final constraints: curvilinear coordinates and droplet-laser distance.



(a) *Coordinates.* **(b)** *Control.*

Figure 4.10: Reference tracking with no final constraints: coordinates and reconstructed robot control.

4.5 MPC with FE and direct strategy

This Section deals with the reference tracking OCP, Problem TMPC, formulated in §4.1. The *direct strategy* consists in the full discretization of the space-time domain. Here we adopt a FE discretization to get rid of the spatial coordinates, and obtaining a (large-scale) system of ODEs. Time discretization and control parametrization are then needed to numerically evaluate the evolution of the dynamical system. The strategy is graphically resumed in Fig. 4.11.

As proposed in §4.4, a particular control parametrization and a direct search optimizer are adopted to approximate an optimal control. Given a control input and the actual state, and based on its mathematical model, the evolution of the system within a finite time horizon is predicted, and this allows to evaluate a suitable cost functional. This cost is a metric to optimize the injected control input, and a direct search method carries out this task.

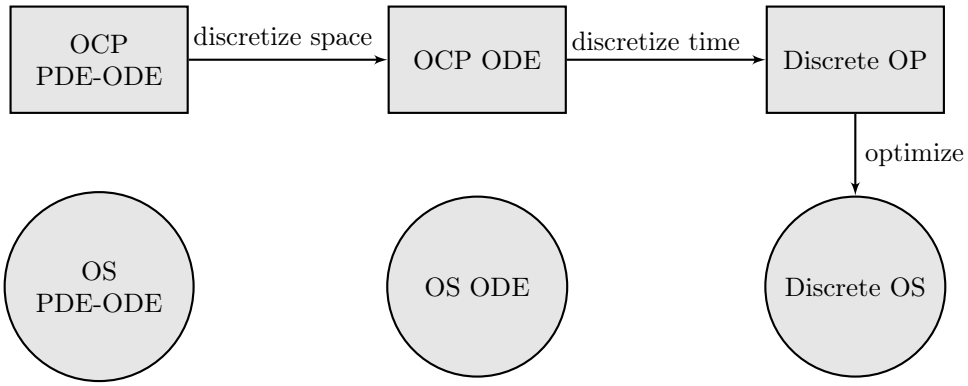


Figure 4.11: Flow chart of the *direct strategy*.

Dynamical model Let us recall the mathematical models of robot (2.5), droplet (2.92) and thermal dynamics (2.69), as developed in Chapter 2, the kinematics of curvilinear coordinates (4.34) and consider the time interval $[0, T]$.

Section 2.3 explains the assumption of constant homogeneous boundary conditions for the temperature field, $\mathbf{w}_\mathrm{c}(t) = \mathbf{1}\bar{w}\ \forall t \in [0, T]$. It is possible to take w.l.o.g. $\bar{w} = 0\,°\mathrm{C}$, so that the equation of thermal dynamics simplifies. Moreover, control input is here composed by motor currents $\mathbf{u}^i$ and laser power u^p, hence $\mathbf{u} = [\mathbf{u}^i, u^p]^T$. The overall model for $t \in [0, T]$ reads:

$$\begin{aligned} \boldsymbol{M}_y \dot{\mathbf{y}}(t) &= \boldsymbol{A}_y\, \mathbf{y}(t) + \boldsymbol{B}_y \mathbf{u}^i(t) \\ \boldsymbol{M}_{\mathrm{ff}} \dot{\mathbf{w}}_\mathrm{f}(t) &= -\boldsymbol{K}_{\mathrm{ff}} \mathbf{w}_\mathrm{f}(t) + \mathbf{f}_\mathrm{f}(\mathbf{y}(t)) u^p(t) \\ \boldsymbol{M}_z \dot{\mathbf{z}}(t) &= \boldsymbol{A}_z(\mathbf{z}(t))\, \mathbf{z}(t) + \boldsymbol{B}_z(\mathbf{z}(t))\, \mathbf{w}(t) \\ \dot{\boldsymbol{\xi}}(t) &= \boldsymbol{R}_\gamma(\boldsymbol{\xi}(t))\, \mathbf{z}(t) \end{aligned} \tag{4.42}$$

Time integration Evolution of the continuous-time system can be approximated by using a discrete-time model of its dynamics. It is worth to introduce a time grid $\Gamma = \{t_k | t_k = k\Delta t, k = 0, 1, \ldots, N\}$, equispaced for simplicity, with time steps size $\Delta t = T/N > 0$. Robot and liquid temperature discrete-time dynamics can be found

considering the trapezoidal rule, as reported in §3.3 and §2.3.2 respectively. Then, an approximated trapezoidal rule can be used for droplet dynamics, in order to avoid an implicit scheme, as shown in §3.4.

$$\left[\boldsymbol{M}_y - \frac{\Delta t}{2}\boldsymbol{A}_y\right]\mathbf{y}_k = \left[\boldsymbol{M}_y + \frac{\Delta t}{2}\boldsymbol{A}_y\right]\mathbf{y}_{k-1} + \Delta t\boldsymbol{B}_y\frac{\mathbf{u}_k^i + \mathbf{u}_{k-1}^i}{2} \tag{4.43}$$

$$\left[\boldsymbol{M}_{\mathrm{ff}} + \frac{\Delta t}{2}\boldsymbol{K}_{\mathrm{ff}}\right]\mathbf{w}_{\mathrm{f},k} = \left[\boldsymbol{M}_{\mathrm{ff}} - \frac{\Delta t}{2}\boldsymbol{K}_{\mathrm{ff}}\right]\mathbf{w}_{\mathrm{f},k-1} + \Delta t\frac{\mathbf{f}_{\mathrm{f}}(\mathbf{y}_k)u_k^p + \mathbf{f}_{\mathrm{f}}(\mathbf{y}_{k-1})u_{k-1}^p}{2} \tag{4.44}$$

$$\left[\boldsymbol{M}_z - \frac{\Delta t}{2}\boldsymbol{A}_z(\mathbf{z}_{k-1})\right]\mathbf{z}_k = \left[\boldsymbol{M}_z + \frac{\Delta t}{2}\boldsymbol{A}_z(\mathbf{z}_{k-1})\right]\mathbf{z}_{k-1} + \Delta t\boldsymbol{B}_z(\mathbf{z}_{k-1})\frac{\mathbf{w}_k + \mathbf{w}_{k-1}}{2} \tag{4.45}$$

Curvilinear coordinates can be updated by applying Heun's method, as developed in §4.4.

$$\hat{\boldsymbol{\xi}}_k = \boldsymbol{\xi}_{k-1} + \Delta t\boldsymbol{R}_\gamma(\boldsymbol{\xi}_{k-1})\mathbf{z}_{k-1} \tag{4.46}$$

$$\boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1} + \Delta t\frac{\boldsymbol{R}_\gamma(\hat{\boldsymbol{\xi}}_k)\mathbf{z}_k + \boldsymbol{R}_\gamma(\boldsymbol{\xi}_{k-1})\mathbf{z}_{k-1}}{2} \tag{4.47}$$

Control and constraints In order to speed up the optimization process, perhaps obtaining less accurate solutions, a suitable control parametrization is needed. There exist different possibilities to obtain a low dimensional optimization space by mapping optimization variables and controls at different levels. On example is the position control proposed in §4.4.1, where N_L^* laser pulses are emitted on the same position $\mathbf{x}_L^*$. Other approaches comprise, for instance, the parametrization of robot currents, acceleration or velocity, or robot path through the coefficients of parametric curves. However, the main idea is to keep as small as possible the optimization space in order to bound the computational time needed to find an (approximated) optimal control.

Constraints on control inputs may account for physical limitations of the system and may be embedded into the optimization problem to further reduce the search space. Actually, how the constraints are formulated depend on the specific control parametrization. For instance, robot displacement during one time step is bounded in an interval that depends on the actual state of the robot $\mathbf{y}$ and on the available control input to the robot (e. g. motor currents $-\mathbf{i}^{\max} \leq \mathbf{i} \leq \mathbf{i}^{\max}$). The constraint on the next robot position may be $\mathbf{y}_k^{\min} \leq \mathbf{y}_k \leq \mathbf{y}_k^{\max}$ where bounds $\mathbf{y}_k^{\min}$ and $\mathbf{y}_k^{\max}$ are given by:

$$\begin{aligned}\left[\boldsymbol{M}_y - \frac{\Delta t}{2}\boldsymbol{A}_y\right]\mathbf{y}_k^{\max} &= \left[\boldsymbol{M}_y + \frac{\Delta t}{2}\boldsymbol{A}_y\right]\mathbf{y}_{k-1} + \Delta t\boldsymbol{B}_y\frac{\mathbf{i}^{\max} + \mathbf{u}_{k-1}^i}{2}\\ \left[\boldsymbol{M}_y - \frac{\Delta t}{2}\boldsymbol{A}_y\right]\mathbf{y}_k^{\min} &= \left[\boldsymbol{M}_y + \frac{\Delta t}{2}\boldsymbol{A}_y\right]\mathbf{y}_{k-1} + \Delta t\boldsymbol{B}_y\frac{-\mathbf{i}^{\max} + \mathbf{u}_{k-1}^i}{2}\end{aligned} \tag{4.48}$$

Cost functional Cost functional $\mathcal{J}_{\mathrm{MPC}}$ is here considered, as in §4.4. Given time grid Γ, relative weight ω_n and curvilinear coordinates $\boldsymbol{\xi}_k = [s_k, n_k]^T \approx \boldsymbol{\xi}(t_k)$ for $t_k \in \Gamma$, its discrete-time approximation is given in Eq. (4.39), namely:

$$\mathcal{J}_{\mathrm{MPC}} \approx -s_N + \omega_n\frac{\Delta t}{2}\sum_{k=0}^{N-1}\left(n_k^2 + n_{k+1}^2\right) \tag{4.49}$$

It is worth to remark that control effort is not penalised due to the simple definition of these costs.

MPC setup Model predictive control permits to compute the next control input to optimally drive the system. Based on the actual state of the system and the discrete-time model (4.53), the optimal control can be found by minimizing the cost functional (4.49). This task has to be repeated with a given rate (period T_{MPC}) in order to compensate for disturbances and modelling inaccuracies [3]; notice that the rate may depend on system state instead of on time explicitly.

Evaluation of the cost functional requires to have the evolution of the system, typically within a finite time horizon $T_{\text{horiz}} > T_{\text{MPC}}$. Thus, at time t^* the time interval $[t^*, \min(t^* + T_{\text{horiz}}, T)]$ is considered by the model-based controller. The mathematical model of the system, the actual state $[\mathbf{y}(t^*), \mathbf{w}(t^*), \mathbf{z}(t^*), \boldsymbol{\xi}(t^*)]^T$ and a time grid with constant step size Δt_{MPC} lead to the evolution of the system for any given injected control—$N_{\text{MPC}} = T_{\text{horiz}}/\Delta t_{\text{MPC}}$ updates of the discrete-time dynamics are needed for each MPC run. Then, the cost functional (4.49) can be computed and optimized. The optimal control is fed into the system, that evolves accordingly (discrete-time dynamics with time step size Δt); after a period T_{MPC}, MPC runs again starting from $t^* + T_{\text{MPC}}$ and the updated state of the system.

In order to reduce the optimization space and to lighten the problem, the control is redefined as robot position and laser power, $\mathbf{u} = [\mathbf{u}^x, u^p]^T$, instead of robot currents and laser power. So, robot dynamics is no more included in the MPC problem. Moreover, the control is parametrized with one laser position only and the laser power profile is fixed. In particular, given a position $\mathbf{x}_L$ the control can be written in continuous-time for $t \in [0, T_{\text{horiz}}]$ as:

$$\mathbf{u}^x(t) = \mathbf{x}_L \tag{4.50}$$

$$u^p(t) = \begin{cases} p_L & t \le T_{\text{MPC}} \\ 0 & \text{otherwise} \end{cases} \tag{4.51}$$

This means that the OCP (4.7) is converted into an NLP with two optimization variables only. The dynamical model cited above changes and reduces to:

$$\begin{aligned} \boldsymbol{M}_{\text{ff}}\dot{\mathbf{w}}_{\text{f}}(t) &= -\boldsymbol{K}_{\text{ff}}\mathbf{w}_{\text{f}}(t) + \mathbf{f}_{\text{f}}(\mathbf{x}_L)u^p(t) \\ \boldsymbol{M}_z\dot{\mathbf{z}}(t) &= \boldsymbol{A}_z(\mathbf{z}(t))\,\mathbf{z}(t) + \boldsymbol{B}_z(\mathbf{z}(t))\,\mathbf{w}(t) \\ \dot{\boldsymbol{\xi}}(t) &= \boldsymbol{R}_\gamma(\boldsymbol{\xi}(t))\,\mathbf{z}(t) \end{aligned} \tag{4.52}$$

where laser power profile $u^p = u^p(t)$ is given by (4.51).

Let us denote the system state $\boldsymbol{\chi} = [\mathbf{w}, \mathbf{z}]^T$. As explained above, discrete-time update equations can be derived by using trapezoidal rule and Heun's method. For the sake of simplicity, these expressions can be condensed into the following:

$$\begin{aligned} \boldsymbol{\chi}_k &= \boldsymbol{\psi}(\boldsymbol{\chi}_{k-1}, \mathbf{x}_L, u^p_{k-1,k}) \\ \boldsymbol{\xi}_k &= \boldsymbol{\varphi}(\boldsymbol{\xi}_{k-1}, \boldsymbol{\chi}_{k-1,k}) \end{aligned} \tag{4.53}$$

Finally, the MPC problem at time $t_j \in \Gamma$, $t_j = j\Delta t$, can be expressed as:

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}_{\text{MPC}} \\ \text{subject to} \quad & \boldsymbol{\chi}_k = \boldsymbol{\psi}(\boldsymbol{\chi}_{k-1}, \mathbf{x}_L, u^p_{k-1,k}) \\ & \boldsymbol{\xi}_k = \boldsymbol{\varphi}(\boldsymbol{\xi}_{k-1}, \boldsymbol{\chi}_{k-1,k}) \\ & \boldsymbol{\chi}_0 = \hat{\boldsymbol{\chi}}_j \\ & \boldsymbol{\xi}_0 = \hat{\boldsymbol{\xi}}_j \\ \text{for} \quad & k \in \{1, \dots, N_{\text{MPC}}\} \end{aligned} \tag{4.54}$$

where initial conditions are given by the system evolution; in particular they are $\hat{\boldsymbol{\chi}}_j \approx \boldsymbol{\chi}(t_j)$ and $\hat{\boldsymbol{\xi}}_j \approx \boldsymbol{\xi}(t_j)$. Section 4.5.1 discusses about numerical methods to face this problem and to find a solution, perhaps approximated.

Time horizon T_{horiz} may have a great impact on the optimal control. For instance, how car drivers control the vehicle may depend on the distance of sight, the presence of corners or the like. An optimal time horizon should be adaptively selected to take into account the control problem and the reference signal [81]. Characteristic time constants of the system may suggest a lower bound for the time horizon. Evaluating the thermal time constant, with characteristic length $l = 1 - 3\,\text{mm}$, it is $\tau_w = l^2/\alpha \approx 7 - 62.6\,\text{s}$. Dynamics of droplet migration is almost in the same range, see §3.2.1. However, the computational effort to let the system evolve for T_{horiz} is high, due to the small time step size Δt_{MPC} to have an accurate evolution.

Let us discuss about the relationship between computational time and MPC setup. Denoting Δt_{int} the time needed to update one step of the discrete-time evolution, then the time to compute the overall evolution in T_{horiz} is almost $\Delta t_{\text{int}} N_{\text{MPC}}$. The evaluation of the cost functional is called several times by the optimizer; this number of iterations may be upper bounded by $N_{\max}$. Thus, considering the time between two runs of the MPC, T_{MPC}, it has to be long enough to let the optimizer find a solution. This leads to a constraint on the maximum allowed integration time Δt_{int}, namely:

$$\begin{aligned} T_{\text{MPC}} &\geq \Delta t_{\text{int}} N_{\text{MPC}} N_{\max} \\ &\Downarrow \\ \Delta t_{\text{int}} &\leq \frac{T_{\text{MPC}}}{N_{\max} N_{\text{MPC}}} \end{aligned} \tag{4.55}$$

Desirable values of T_{MPC} and Δt_{MPC} are small in general, larger $N_{\max}$ typically returns a more accurate solution and most of the times it is $T_{\text{horiz}} \gg T_{\text{MPC}}$. As an example, given $T_{\text{MPC}} = 50\,\text{ms}$, $\Delta t_{\text{MPC}} = 10\,\text{ms}$ and $T_{\text{horiz}} = 2\,\text{s}$, and therefore $N_{\text{MPC}} = 200$, the maximum integration time has to be $\Delta t_{\text{int}} = 5 - 25\,\mu\text{s}$ to limit the optimizer at $N_{\max} = 10 - 50$ iterations.

Robot controller The MPC problem formulated above, Problem (4.54), does not include robot dynamics. However, the dynamics of the robot can affect the evolution of the system, and its influence depends on how fast it is w.r.t. droplet and liquid temperature dynamics. Let us denote $\mathbf{x}_k^*$ the solution of (4.54) at time t_{k-1}, that is (approximately) the optimal position of the laser beam at time t_k. Thus, position $\mathbf{x}_k^*$ can be used as a reference to drive the robot.

Here a simple and effective control scheme is described. A proportional feedback control can be used to select the next robot velocity $\mathbf{v}_k^* = \kappa(\mathbf{x}_k^* - \mathbf{y}_{k-1}^x)$, that is actually a reference because the robot is controlled by motor currents; $\mathbf{y}_{k-1}^x$ is the actual robot position. Notice that selecting the feedback gain $\kappa = 1/\Delta t > 0$, the feedback law resembles the application of backward Euler scheme.

$$\mathbf{x}_k^* = \mathbf{y}_{k-1}^x + \mathbf{v}_k^* \Delta t \qquad \Leftrightarrow \qquad \mathbf{v}_k^* = \frac{1}{\Delta t}\left(\mathbf{x}_k^* - \mathbf{y}_{k-1}^x\right) \tag{4.56}$$

Then, motor currents needed to reach the target velocity $\mathbf{v}_k^*$ can be computed by inverting the state update equation derived with the trapezoidal rule (3.25). This leads

to an expression of target motor currents $\mathbf{i}_k^*$, namely:

$$\begin{gathered}\left(\frac{m_L}{\Delta t}+\frac{c_L}{2}\right)\mathbf{v}_k^* = \left(\frac{m_L}{\Delta t}-\frac{c_L}{2}\right)\mathbf{y}_{k-1}^v + b_L\frac{\mathbf{i}_{k-1}+\mathbf{i}_k^*}{2} \\ \Downarrow \\ \mathbf{i}_k^* = \frac{2}{b_L}\left[\left(\frac{m_L}{\Delta t}+\frac{c_L}{2}\right)\kappa\left(\mathbf{x}_k^*-\mathbf{y}_{k-1}^x\right)-\left(\frac{m_L}{\Delta t}-\frac{c_L}{2}\right)\mathbf{y}_{k-1}^v\right]-\mathbf{i}_{k-1}\end{gathered} \tag{4.57}$$

Finally, motor currents $\mathbf{i}_k^*$ are saturated at feasible bounds and applied, then the robot state evolves according to its dynamic model.

4.5.1 Numerical solution

The direct single shooting method is here adopted to deal with the MPC problem formulated above. This means that the system evolution is computed, then the cost functional is evaluated and fed into an optimizer that changes the control input to decrease the value of the cost. Doing so, state and control constraints are always satisfied.

It has to be underlined the advantage of Cholesky decomposition of matrices related to the FE discretization. This is useful because they are constant (thanks to the assumptions on constant homogeneous physical properties and linear thermal behaviour of the liquid) and especially because they are large. Moreover, FE matrices are sparse and this can be exploited by adopting suitable permutations, e. g. minimum-degree pivoting, to maintain the sparsity in the decomposition.

Optimizer A direct search optimizer is adopted to find the control that minimizes the cost functional, solving (perhaps approximately) Problem (4.54). Optimization for nonlinear MPC through a direct search method has been recently discussed in [78]. In particular, the `patternsearch` routine provided by Matlab is used. This allows to impose constraints and bounds on the optimization variables, as suggested above to take into account robot dynamics.

Gradient-based optimization has been widely developed and applied in the solution of non-linear OCPs. Different methods could be adopted, e. g. shooting, collocation and direct discretization methods. In the case of linear systems gradient-based methods are typically fast [5], but this may occur for non-linear MPC too [39].

4.5.2 Simulation results

Tests The approach has been tested with the reference path used in §4.4, free final conditions and cost functional $\mathcal{J}_{\text{MPC}}$ (4.6). Moreover, presence or not of robot dynamics in the system has been simulated. Control performances are expected to reduce when robot dynamics appears in the control loop, because it is a disturbance from the controller point-of-view. This test may highlight the need of accounting for robot dynamics in the formulation of the MPC problem.

Liquid and droplet properties and reference robot parameters are reported in Tab. 3.2 and Tab. 4.1 respectively. The liquid layer has been discretized with 4-nodes rectangular finite elements and the temperature field is interpolated using piece-wise linear shape functions, Tab. 4.5. Different time steps have been selected for evolution and simulated evolution in MPC, $\Delta t < \Delta t_{\text{MPC}}$, in order to spare computational time,

Table 4.5: FE discretization of the liquid layer.

Domain Ω	$[0, 16] \times [0, 12]$	mm × mm
FE size	0.3×0.3	mm × mm
FE nodes	4	

Table 4.6: MPC parameters.

T_{horiz}	1	s
Δt	0.01	s
T_{MPC}	0.03	s
Δt_{MPC}	0.03	s
ω_n	3.33×10^4	$(\text{m s})^{-1}$

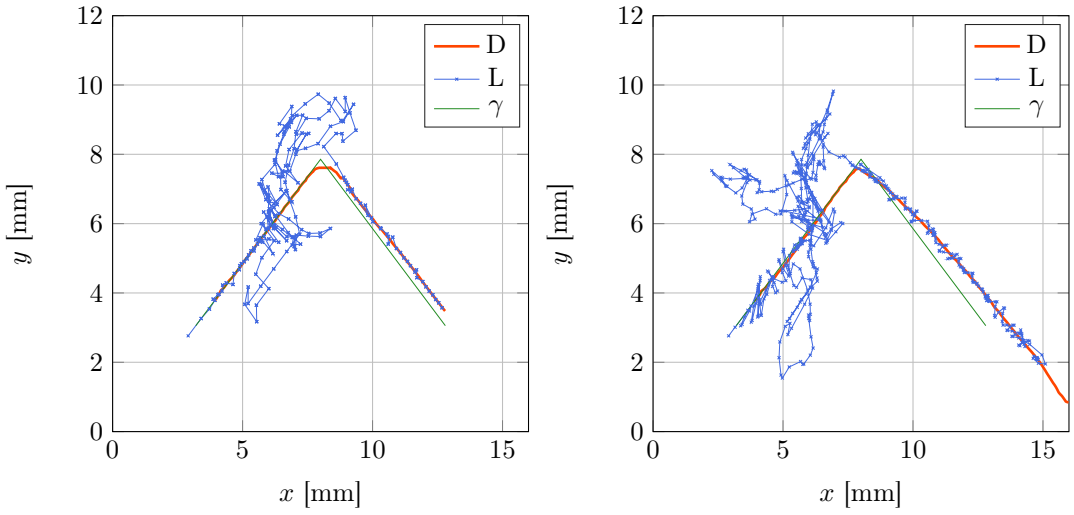
perhaps at the cost of a less accurate model-based control action. The relative weight ω_n in the cost functional is the same adopted in §4.4. Options reported in Tab. 4.7 are chosen for the direct search optimizer `patternsearch` provided by Matlab. Initial droplet position is given, while robot is placed close to the droplet by the user or according to the solution of an MPC run. The initial guess of each optimization process may be the actual laser or droplet position. The latter works better and seems to be more robust; it is used in these simulations. Moreover, the optimal laser position is constrained to be in a rectangular domain, defined considering the robot dynamics. Lower and upper bounds have been computed with Eq. (4.48) at each time step.

Results and discussion Let us discuss about the evolution of the system with the MPC loop and instantaneous robot positioning. Notice that laser and droplet paths and the time profile of droplet curvilinear coordinates resemble the results obtained in §4.4 with the *analytical strategy*. In particular, compare Fig. 4.12a and Fig. 4.13a to Fig. 4.8 and Fig. 4.9a. The system evolution, and then the underlying control, is almost the same. This happens because the two dynamical models represent the same physical system and the cost functional is the same. Little differences may arise because of the different time horizon T_{horiz} or accuracy of the discrete-time evolution.

As before in §4.4, the maximum lateral deviation is small, $\max |n| < 1\,\text{mm}$, and close to the corner, where it is difficult—perhaps impossible—to stay on the reference path. As shown in Fig. 4.13b, during pushing phases, i. e. approximately $t < 0.5\,\text{s}$ and $t > 3\,\text{s}$, the distance laser-droplet is almost constant and its value is similar to those found in §4.3 and §4.4. Considerations about time horizon and control parametrization

Table 4.7: Options for `patternsearch` in Matlab.

MaxIter	200
MaxFunEvals	200
TolX	0.05 mm
CompletePoll	*on*
CompleteSearch	*on*



(a) *No robot.* **(b)** *Robot.*

Figure 4.12: Droplet and laser path with and without robot dynamics.

discussed in §4.4 still hold.

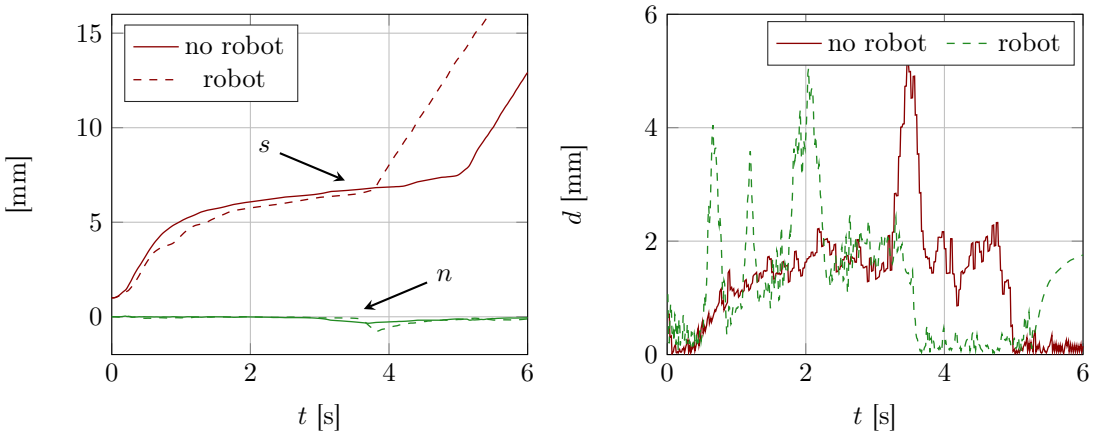
The diagram in Fig. 4.14 shows the relationship between laser-droplet distance and droplet velocity $\|\mathbf{z}^v\|$. Actually, this relation is mediated by the temperature field and its dynamics, and so the diagram does not show a simple function, but highlights the hysteresis of the system. It has to be noticed that the maximum velocity reached by the droplet is around $\|\mathbf{z}^v\| \approx 8\,\text{mm/s}$, with a corresponding distance $d = \|\mathbf{z}^x - \mathbf{y}^x\| \approx 0.07\,\text{mm}$ (recall that droplet is a point-mass).

Evolution of the system and control strategy do not change significantly in the case robot moves with its own dynamics, not instantaneously. As evidenced in Fig. 4.15b, the deviation between desired and actual laser position is on the order of $e = \|\mathbf{y}^x - \mathbf{x}^*\| \approx 0.3\,\text{mm}$, that is comparable with the characteristic radius of the laser beam. However, with little differences, most of the features discussed for the instantaneous robot positioning hold in this case too. Results are graphically reported in Fig. 4.12b and Fig. 4.13.

At about $t = 5.3\,\text{s}$ the robot is stopped by the user and then the rest of the system evolves accordingly. Temperature increases near the laser spot but temperature gradient in droplet position decreases—because it is moving far away—, so the driving force reduces and the droplet slows down due to the drag force. This braking phase is clearly seen in Fig. 4.13b for $t > 5.3\,\text{s}$.

Results of the tracking control of robot reference position are graphically reported in Fig. 4.15. Bounds due to robot dynamics used to constrained the optimal solution are depicted.

A relevant difference w.r.t. the case of instantaneous robot positioning is the overall effectiveness, in terms of speed and tracking. Analysing Fig. 4.13a, one can see that the droplet moves 3 mm more along the reference path in the same time $T = 6\,\text{s}$, while maintaining $\max|n(t)| < 1\,\text{mm}$, $t \in [0, T]$. This is not due to higher velocities reached by the droplet, whose peak is actually lower, at about $\|\mathbf{z}^v\| \approx 7\,\text{mm/s}$. On the other



(a) *Curvilinear coordinates.* **(b)** *Droplet-laser distance.*

Figure 4.13: Curvilinear coordinates and droplet-laser distance with and without robot dynamics.

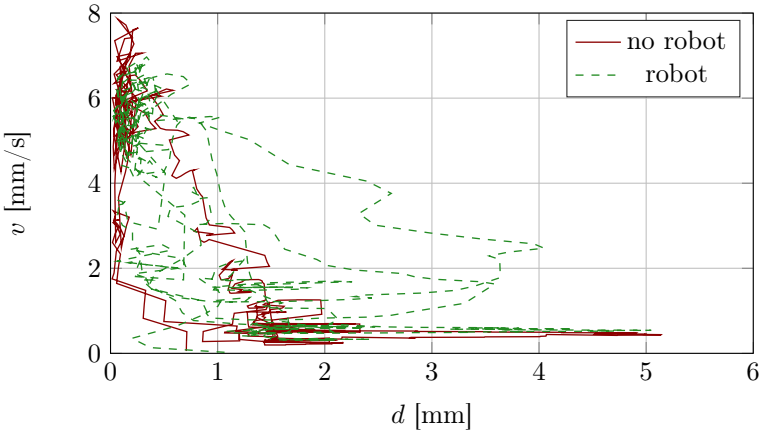
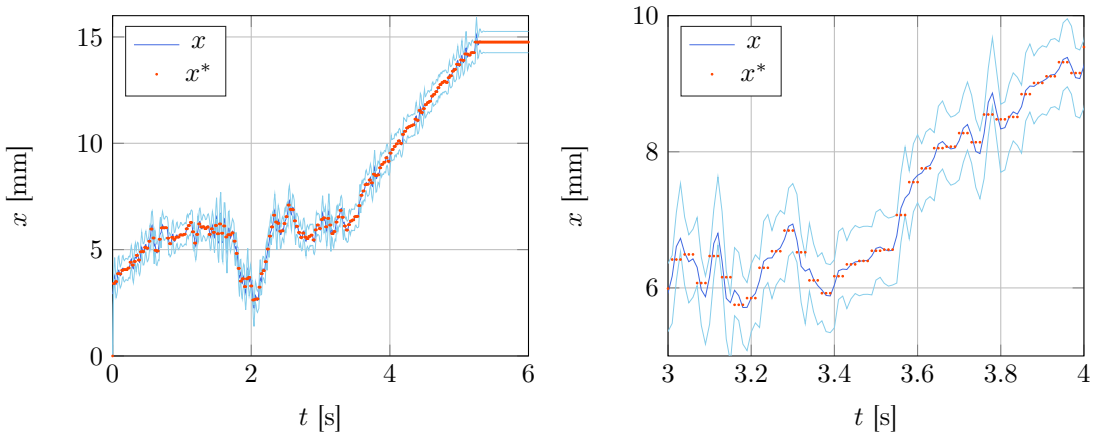


Figure 4.14: Relationship between droplet-laser distance and droplet velocity.



(a) *Position along x axis.* **(b)** *Zoom.*

Figure 4.15: Tracking of reference laser position and bounds with robot dynamics.

hand, Fig. 4.14 shows that points corresponding to robot dynamics spread to a larger area. This means that the system dynamics is—surprisingly—more exploited. A deeper investigation is needed to better understand the impact of robot dynamics in the MPC closed-loop.

Conclusions

The work started with the definition of the elements involved in the system and needed to describe its behaviour. A set of mathematical models have been proposed to describe a robot, the laser-liquid interaction, the thermal behaviour of a suspending liquid and the dynamics of a droplet subject to thermocapillary forces. Then, characteristic parameters were identified and model validated thanks to the experimental data collected during this work. The overall model with identified parameters shows good agreement with experimental results for short time intervals ($\Delta t \approx 30\,\mathrm{s}$), until convective flows significantly affect the liquid motion.

Based on this validated (perhaps simple) model, different control strategies have been implemented and tested. Direct discretization method with semi-analytical solution of temperature field was successfully solved both with gradient-based and derivative-free optimizers. Then, model-predictive control with FE model of liquid thermal behaviour was faced with direct method and a direct search optimizer, leading to satisfactory control performances.

Future developments As it often happens in research, the results presented in this thesis can only mark a step within a work in progress. Moreover, the explorative nature of the work opens the doors to a broad range of analyses and experimental activities.

The modelling activity could be reviewed to include more sophisticated dynamics and details, e. g. the physical extension the droplet and the convective flows. These improvements should be based on experimental work carried out to validate and stress the present model. On the other hand, the complexity of the model could increase a lot, considering the fluid dynamics embedded into the problem. Thus, reduced-order modelling techniques could be of growing interest to a suitable control-oriented mathematical model, that has to be accurate enough yet simple.

Many optimization-based control techniques have been not tested within this work. Methods suitable for large-scale problems or methods that use a local approximation of the dynamics may be analysed and implemented to evaluate their performance. In particular, we refer to the real-time iteration scheme [3, 39] and to the differential dynamic programming [7,37,79]. Moreover, indirect methods could be exploited [4,6,76]. There are, to the author's knowledge, applications of indirect methods in OCPs with PDEs only for small-scale problems.

Appendix A

Analytical and Finite Elements Solution

This Appendix compares two methods to model the thermal dynamics of the liquid layer. In particular, in §2.3.1 a semi-analytical method is proposed based on the superposition of a finite series of instantaneous laser pulses. This approach takes advantage of the linearity and availability of an analytical solution for the instantaneous pulse, for both Gaussian and point-wise intensity profile of the laser beam. The second method, proposed in §2.3.2, is based on FE discretization of the liquid domain. This approach has been highly developed [60] and can be applied to every problem governed by PDEs.

Benchmark The analytical solution of the temperature field for a continuous laser beam moving at constant speed is known and derived in §2.3.1. Thus, this scenario can be used to evaluate and then compare the temperature field predicted by different models, namely:

M1 analytical solution for point-wise continuous moving heating source (2.49),

M2 finite sum approximation for point-wise (2.20) and Gaussian (2.44) heating source,

M3 finite element approximation and trapezoidal rule for point-wise and Gaussian heating source (2.71).

The simulations are based on physical properties of the liquid reported in Tab. 3.2; other parameters are given in Tab. A.1.

Some are the limitations for an effective comparison, but their impact may be neglected if only M2 and M3 have to be compared. First of all, it is not possible to create a scenario that exactly reproduces the hypotheses underlying the model M1, in particular the steady state condition from the laser point-of-view.

The following analyses have been carried out:

T1 prediction of models M1 and M2 for point-wise and Gaussian heating source, considering two different time steps, Δt_1 and Δt_2;

T2 prediction of model M3 for Gaussian heating source, considering two different time steps, Δt_1 and Δt_2, and two different characteristic radii, $r_{L,1}$ and $r_{L,2}$;

T3 prediction of models M1, M2 and M3 for point-wise heating source, considering time step Δt,

Table A.1: Simulation parameters.

u	5	mm/s
p_L	0.5	W
β	100	1/m
l_z	1	mm
$r_{L,1}$	0.25	mm
$r_{L,2}$	1	mm
Δt_1	10	ms
Δt_2	60	ms
Δt	40	ms
FE domain	50×20	mm $\times$ mm
FE size	0.2×0.2	mm $\times$ mm
FE nodes	4	

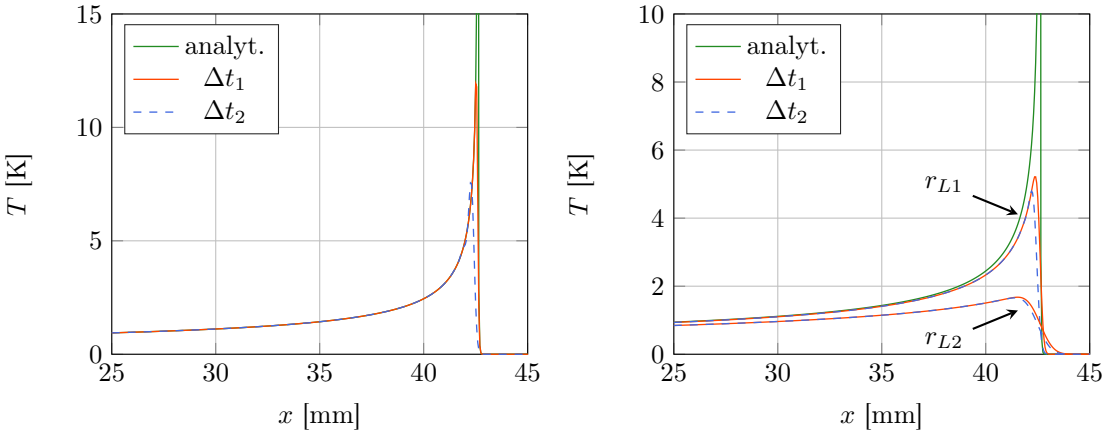
T4 prediction of models M2 and M3 for Gaussian heating source, considering time step Δt and radii $r_{L,1}$ and $r_{L,2}$.

Test T1 gives more insight about the impact of time step in the approximation of the temperature field. In particular, this happens because time step affects the discretization of convolution integral (2.42); convergence is expected as the time step tends to zero. Test T2 is built to analyse the convergence of the thermal evolution with FE discretization, considering the time discretization based on the trapezoidal rule, Eq. (2.71). Then, tests T3 and T4 compare analytical and FE solutions, for both point-wise and Gaussian laser beam profile.

Results and discussion Results of T1 are graphically reported in Fig. A.1. In the case of point-wise laser beam, the temperature peak is almost halved considering the longer time step Δt_2; on the other hand, convergence of models M1 and M2 is clearly seen in Fig. A.1a. In the case of Gaussian laser beam, the temperature profile is more robust to the time discretization, and the error decreases as the characteristic radius increases. Moreover, models M1 and M2 converge as time step and characteristic radius tend to zero, Fig. A.1b.

Convergence of predictions with FEM is evidenced in Fig. A.2. For the larger characteristic radius, $r_{L,2}$, they are almost superimposed. Instead, a small deviation appears for $r_{L,1}$. As observed above, temperature profile is robust to time discretization in the case of Gaussian laser beam. Let us discuss about the prediction of temperature close to the narrower laser beam: a negative temperature difference arises, and it is larger with the shorter time step. We suggest that this phenomenon happens because the spatial discretization is too coarse, in particular the element size is on the order of the characteristic radius, FE size $\approx r_{L,2}$. The same effect is amplified in the case of point-wise laser beam, Fig. A.3a.

Results of tests T3 and T4 are graphically reported in Fig. A.3a and Fig. A.3b. It is clear that values of predicted temperature field do not match between analytical (M2) and FE solution (M3). However, the behaviour is quite the same, especially in the case of Gaussian laser beam.



(a) *Point-wise laser beam.* **(b)** *Gaussian laser beam.*

Figure A.1: Continuous moving laser beam: effect of time step on temperature field predicted with semi-analytical solutions.

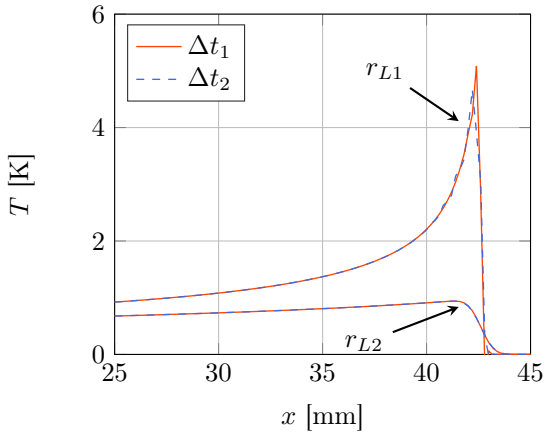
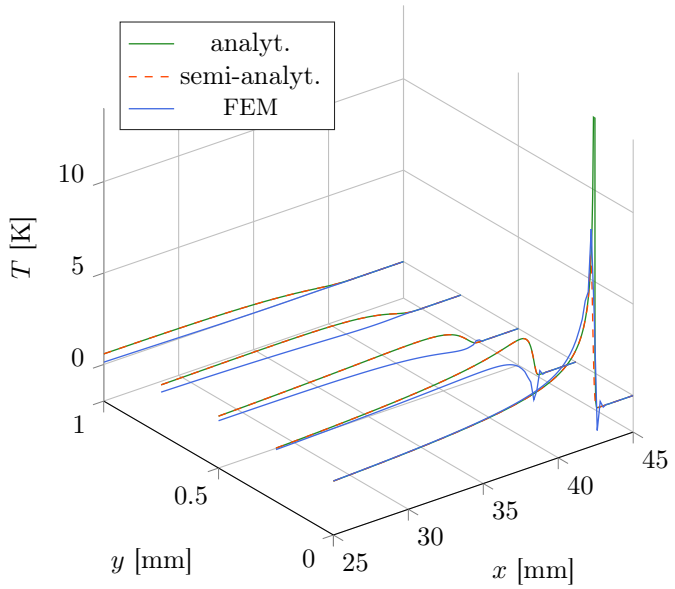
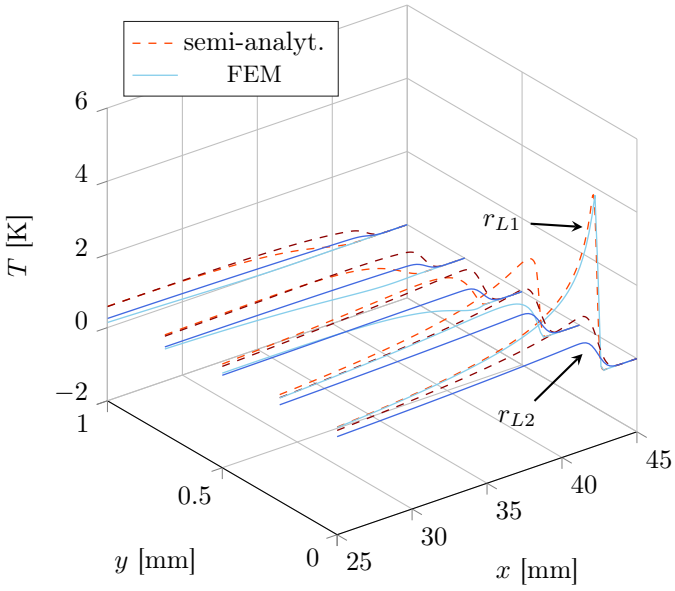


Figure A.2: Continuous moving laser beam: effect of time step on temperature field predicted with FEM.



(a) *Point-wise laser beam.*



(b) *Gaussian laser beam.*

Figure A.3: Comparison of temperature field predicted by semi-analytical solution and FEM.

A deeper analysis is needed to investigate the effect of spatial resolution, laser velocity, physical properties and laser parameters. Moreover, the overall extension of the domain discretized with FE may have an influence on the predicted temperature field, because of the boundary conditions (that maybe tend to cool down the liquid).

Appendix B

Derivation of constraints Jacobian matrix

Let us recall the function of equality constraints $\mathbf{g}$ for the OCP analysed in §4.3, Eq. (4.23). This Appendix deals with the derivation of its Jacobian matrix $\boldsymbol{J}$ w.r.t. optimization variables $\mathbf{x}$. For the sake of simplicity, we take $\mathbf{u} = \mathbf{u}^i$ and constant laser power $u^p = p_L$, thus $n_u = 2$ and function $\boldsymbol{\psi}_\tau$ depends only on the last control input. Moreover, an equispaced time grid is considered, with time step $\Delta t = T/N = T\Delta\tau$.

$$\mathbf{x} = [T, \boldsymbol{\chi}_0, \dots, \boldsymbol{\chi}_N, \mathbf{u}_0, \dots, \mathbf{u}_N]^T \tag{B.1}$$

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\chi}_0 - \boldsymbol{\chi}^i \\ \boldsymbol{B}_T\boldsymbol{\chi}_N - \boldsymbol{\chi}_B^f \\ \boldsymbol{\chi}_1 - \boldsymbol{\chi}_0 - \Delta\tau\, \boldsymbol{\psi}_\tau(\tau_0, T, \boldsymbol{\chi}_0, \mathbf{u}_0) \\ \boldsymbol{\chi}_2 - \boldsymbol{\chi}_1 - \Delta\tau\, \boldsymbol{\psi}_\tau(\tau_1, T, \boldsymbol{\chi}_{0,1}, \mathbf{u}_1) \\ \vdots \\ \boldsymbol{\chi}_N - \boldsymbol{\chi}_{N-1} - \Delta\tau\, \boldsymbol{\psi}_\tau(\tau_{N-1}, T, \boldsymbol{\chi}_{0,\dots,N-1}, \mathbf{u}_{N-1}) \end{bmatrix} \tag{B.2}$$

$$\boldsymbol{J} = \nabla_{\mathbf{x}}\mathbf{g} = \begin{bmatrix} \dfrac{\partial \mathbf{g}}{\partial T} & \nabla_{\boldsymbol{\chi}_0}\mathbf{g} & \dots & \nabla_{\boldsymbol{\chi}_N}\mathbf{g} & \nabla_{\mathbf{u}_0}\mathbf{g} & \dots & \nabla_{\mathbf{u}_N}\mathbf{g} \end{bmatrix} \tag{B.3}$$

Let us analyse each entry of $\boldsymbol{J}$ to find explicitly the analytical expression. Matrix $\boldsymbol{J}$ is expected to be not sparse but lower triangular, because of the functional $\boldsymbol{\psi}_\tau$. in the following the entries of $\boldsymbol{J}$ are equal to zero where not specified.

Constraint on the initial condition, $\mathbf{g}_1 = \boldsymbol{\chi}_0 - \boldsymbol{\chi}^i$, depends only on the first state $\boldsymbol{\chi}_0$. Similarly, constraint on the final condition, $\mathbf{g}_2 = \boldsymbol{B}_T\boldsymbol{\chi}_N - \boldsymbol{\chi}_B^f$, depends only on the final state, $\boldsymbol{\chi}_N$.

$$\frac{\partial \mathbf{g}_1}{\partial \boldsymbol{\chi}_1}(\mathbf{x}) = \boldsymbol{I} \tag{B.4}$$

$$\frac{\partial \mathbf{g}_2}{\partial \boldsymbol{\chi}_N}(\mathbf{x}) = \boldsymbol{B}_T \tag{B.5}$$

Then, constraints due to the dynamical model are given by:

$$\mathbf{g}_{k+3}(\mathbf{x}) = \boldsymbol{\chi}_{k+1} - \boldsymbol{\chi}_k - \Delta\tau\, \boldsymbol{\psi}_\tau(\tau_k, T, \boldsymbol{\chi}_{0,\dots,k}, \mathbf{u}_k) \tag{B.6}$$

for $k = 0, 1, \ldots, N-1$. Partial derivatives of $\mathbf{g}_{k+3}$ w.r.t. optimization variables $\mathbf{x}$ are easily obtained.

$$\begin{aligned}
\frac{\partial \mathbf{g}_{k+3}}{\partial T}(\mathbf{x}) &= -\Delta\tau \frac{\partial \boldsymbol{\psi}_\tau}{\partial T}(\tau_k, T, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_k) \\
\frac{\partial \mathbf{g}_{k+3}}{\partial \boldsymbol{\chi}_j}(\mathbf{x}) &= \begin{cases} -\Delta\tau \frac{\partial \boldsymbol{\psi}_\tau}{\partial \boldsymbol{\chi}_j}(\tau_k, T, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_k) & j = 0, \ldots, k-1 \\ -\boldsymbol{I} - \Delta\tau \frac{\partial \boldsymbol{\psi}_\tau}{\partial \boldsymbol{\chi}_j}(\tau_k, T, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_k) & j = k \\ \boldsymbol{I} & j = k+1 \end{cases} \\
\frac{\partial \mathbf{g}_{k+3}}{\partial \mathbf{u}_j}(\mathbf{x}) &= \left\{ -\Delta\tau \frac{\partial \boldsymbol{\psi}_\tau}{\partial \mathbf{u}_j}(\tau_k, T, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_k) \qquad j = k \right.
\end{aligned} \tag{B.7}$$

Let us focus on the derivatives of function $\boldsymbol{\psi}_\tau$, whose expression is given in Eq. (4.17). Let us write the state vector as $\boldsymbol{\chi} = [\mathbf{x}_D, \mathbf{v}_D, \mathbf{x}_L, \mathbf{v}_L]^T$. Then function $\boldsymbol{\psi}_\tau$ is:

$$\boldsymbol{\psi}_\tau(\tau_k, T, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_k) = T \begin{bmatrix} \mathbf{v}_{D,k} \\ \frac{b_D}{m_D}\mathbf{g}_w(\tau_k, T, \mathbf{x}_{D,k}, \mathbf{x}_{L,0,\ldots,k}) - \frac{c_D}{m_D}\|\mathbf{v}_{D,k}\|\mathbf{v}_{D,k} \\ \mathbf{v}_{L,k} \\ \frac{b_L}{m_L}\mathbf{u}_k - \frac{c_L}{m_L}\mathbf{v}_{L,k} \end{bmatrix} \tag{B.8}$$

Denoting $(\cdot) = (\tau_k, T, \boldsymbol{\chi}_{0,\ldots,k}, \mathbf{u}_k)$, partial derivative of $\boldsymbol{\psi}_\tau$ w.r.t. final time T is:

$$\frac{\partial \boldsymbol{\psi}_\tau}{\partial T}(\cdot) = \begin{bmatrix} \mathbf{v}_{D,k} \\ T\frac{b_D}{m_D}\frac{\partial \mathbf{g}_w}{\partial T}(\cdot) + \frac{b_D}{m_D}\mathbf{g}_w(\cdot) - \frac{c_D}{m_D}\|\mathbf{v}_{D,k}\|\mathbf{v}_{D,k} \\ \mathbf{v}_{L,k} \\ \frac{b_L}{m_L}\mathbf{u}_k - \frac{c_L}{m_L}\mathbf{v}_{L,k} \end{bmatrix} \tag{B.9}$$

Partial derivative of $\boldsymbol{\psi}_\tau$ w.r.t. state $\boldsymbol{\chi}_k$ is:

$$\frac{\partial \boldsymbol{\psi}_\tau}{\partial \boldsymbol{\chi}_k}(\cdot) = T \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \frac{b_D}{m_D}\frac{\partial \mathbf{g}_w}{\partial \mathbf{x}_{D,k}}(\cdot) & -2\frac{c_D}{m_D}\|\mathbf{v}_{D,k}\|\boldsymbol{I} & \frac{b_D}{m_D}\frac{\partial \mathbf{g}_w}{\partial \mathbf{x}_{L,k}}(\cdot) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{c_L}{m_L}\boldsymbol{I} \end{bmatrix} \tag{B.10}$$

and for $j = 0, \ldots, k-1$ it simplifies because only past laser positions $\mathbf{x}_{L,j}$ have an impact, namely:

$$\frac{\partial \boldsymbol{\psi}_\tau}{\partial \boldsymbol{\chi}_j}(\cdot) = T \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{b_D}{m_D}\frac{\partial \mathbf{g}_w}{\partial \mathbf{x}_{L,j}}(\cdot) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{B.11}$$

Partial derivative of $\boldsymbol{\psi}_\tau$ w.r.t. control $\mathbf{u}_k$ is (assuming $\mathbf{u} = \mathbf{u}^i$):

$$\frac{\partial \boldsymbol{\psi}_\tau}{\partial \mathbf{u}_k} = T \begin{bmatrix} \mathbf{0} \\ \frac{b_L}{m_L}\boldsymbol{I} \end{bmatrix} \tag{B.12}$$

Finally, derivatives of temperature gradient $\mathbf{g}_w$ have to be explicitly computed. Let us collect terms $A_{k,j}$, $B_{k,j}$ and $\delta w_{k,j}$:

$$A_{k,j} = 8\alpha T \Delta\tau (k-j) + r_L^2 \tag{B.13}$$

$$B_{k,j} = 8\alpha T \Delta\tau (k-j) \tag{B.14}$$

$$\delta w_{k,j} = T\Delta\tau \frac{8\beta}{\pi \varrho c_p} \frac{p_{L,j}}{A_{k,j}} \tag{B.15}$$

and rewrite Eq. (2.45) to get:

$$\begin{aligned}\mathbf{g}_w(\cdot) &= -T\Delta\tau\frac{8}{\pi}\frac{\beta}{\varrho c_p}\sum_{j=0}^{k}\frac{p_{L,j}}{A_{k,j}^2}e^{-2\frac{\|\mathbf{x}_{D,k}-\mathbf{x}_{L,j}\|^2}{A_{k,j}}}(\mathbf{x}_{D,k}-\mathbf{x}_{L,j})^T \\ &= -\sum_{j=0}^{k}\frac{\delta w_{k,j}}{A_{k,j}}e^{-\frac{2\|\mathbf{x}_{D,k}-\mathbf{x}_{L,j}\|^2}{A_{k,j}}}(\mathbf{x}_{D,k}-\mathbf{x}_{L,j})^T\end{aligned} \tag{B.16}$$

Temperature gradient $\mathbf{g}_w(\cdot)$, i. e. the temperature gradient in droplet position at time $t_k = kT\Delta\tau$, is a function of final time, T, actual droplet position, $\mathbf{x}_{D,k}$, and previous laser beam positions, $\mathbf{x}_{L,j}$ for $j = 0, 1, \ldots, k$; droplet and laser velocities do not matter. Then, partial derivatives of temperature gradient $\mathbf{g}_w$ are given by:

$$\begin{aligned}&\frac{\partial\mathbf{g}_w}{\partial T}(\cdot) = \\ &= -\frac{1}{T}\sum_{j=0}^{k}\frac{\delta w_{k,j}}{A_{k,j}}\left[1+2\frac{B_{k,j}}{A_{k,j}}\left(\frac{\|\mathbf{x}_{D,k}-\mathbf{x}_{L,j}\|^2}{A_{k,j}}-1\right)\right]e^{-\frac{2\|\mathbf{x}_{D,k}-\mathbf{x}_{L,j}\|^2}{A_{k,j}}}(\mathbf{x}_{D,k}-\mathbf{x}_{L,j})^T\end{aligned} \tag{B.17}$$

$$\begin{aligned}&\frac{\partial\mathbf{g}_w}{\partial\mathbf{x}_{D,k}}(\cdot) = \\ &\quad= -\sum_{j=0}^{k}\frac{\delta w_{k,j}}{A_{k,j}}e^{-2\frac{\|\mathbf{x}_{D,k}-\mathbf{x}_{L,j}\|^2}{A_{k,j}}}\left[\boldsymbol{I}-\frac{4}{A_{k,j}}(\mathbf{x}_{D,k}-\mathbf{x}_{L,j})(\mathbf{x}_{D,k}-\mathbf{x}_{L,j})^T\right]\end{aligned} \tag{B.18}$$

$$\frac{\partial\mathbf{g}_w}{\partial\mathbf{x}_{L,j}}(\cdot) = \frac{\delta w_{k,j}}{A_{k,j}}e^{-2\frac{\|\mathbf{x}_{D,k}-\mathbf{x}_{L,j}\|^2}{A_{k,j}}}\left[\boldsymbol{I}-\frac{4}{A_{k,j}}(\mathbf{x}_{D,k}-\mathbf{x}_{L,j})(\mathbf{x}_{D,k}-\mathbf{x}_{L,j})^T\right] \tag{B.19}$$

It is interesting to notice the relationship between (B.18) and (B.19):

$$\frac{\partial\mathbf{g}_w}{\partial\mathbf{x}_{D,k}}(\cdot) = -\sum_{j=0}^{k}\frac{\partial\mathbf{g}_w}{\partial\mathbf{x}_{L,j}}(\cdot) \tag{B.20}$$

This means that an equal translation of droplet and laser positions does not affect the temperature gradient in droplet position, as expected.

$$\frac{\partial\mathbf{g}_w}{\partial\mathbf{x}_{D,k}}(\cdot) + \sum_{j=0}^{k}\frac{\partial\mathbf{g}_w}{\partial\mathbf{x}_{L,j}}(\cdot) = \mathbf{0} \tag{B.21}$$

Moreover, direct calculation of $\frac{\partial\mathbf{g}_w}{\partial\mathbf{x}_{D,k}}(\cdot)$ can be avoided and Eq. (B.20) can be adopted.

Bibliography

[1] M. Murray, D. and J. Yakowitz, S. "Differential dynamic programming and newton's method for discrete optimal control problems," *Journal of Optimization Theory and Applications*, vol. 43, no. 3, pp. 395–414, July 1984, communicated by D. Q. Mayne.

[2] J. R. Banga, R. Irizarry-Rivera, and W. D. Seider, "Stochastic optimization for optimal and model-predictive control," *Computers and Chemical Engineering*, vol. 22, no. 4–5, pp. 603–612, 1998.

[3] M. M. Diehl, "Real–time optimization for large scale nonlinear processes," Ph.D. dissertation, Ruprecht-Karls-Universität Heidelberg, June 2001.

[4] E. Bertolazzi, F. Biral, and M. Da Lio, "Real–time motion planning for multibody systems," *Multibody system dynamics*, vol. 17, pp. 119–139, 2007.

[5] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, March 2010.

[6] T. G. Flaig, "Discretization strategies for optimal control problems with parabolic partial differential equations," Ph.D. dissertation, Universität der Bundeswehr München, 2013.

[7] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.

[8] E. Yakhshi-Tafti, H. J. Cho, and R. Kumar, "Droplet actuation on a liquid layer due to thermocapillary motion: Shape effect," *Applied Physics Letters*, vol. 96, no. 264101, 2010.

[9] S.-Y. Park and P.-Y. Chiou, "Light-Driven Droplet Manipulation Technologies for Lab-on-a-Chip Applications," *Advances in OptoElectronics*, vol. 2011, no. 909174, pp. 1–12, 2011.

[10] M. G. Simon and A. P. Lee, *Microfluidic Droplet Manipulations and Their Applications*. Springer Science+Business Media, 2012, pp. 23–51.

[11] M. M. Hanczyc, "Droplets: Unconventional protocell model with life-like dynamics and room to grow," *Life*, vol. 4, no. 4, pp. 1038–1049, 2014. [Online]. Available: http://www.mdpi.com/2075-1729/4/4/1038

[12] A. S. Basu and Y. B. Gianchandani, "Virtual microfluidic traps, filters, channels and pumps using Marangoni flows," *Journal of Micromechanics and Microengineering*, vol. 18, p. 115031, 2008.

[13] J.-P. Delville, M. Robert De Saint Vincent, R. D. Schroll, H. Chraïbi, B. Issenmann, R. Wunenburger, D. Lasseux, W. W. Zhang, and E. Brasselet, "Laser microfluidics: fluid actuation by light," *Journal of Optics A: Pure and Applied Optics*, vol. 11, pp. 1–31, 2009. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00366706

[14] R. Seemann, M. Brinkmann, T. Pfohl, and S. Herminghaus, "Droplet based microfluidics," *Reports on Progress in Physics*, vol. 75, no. 1, p. 016601, 2012. [Online]. Available: http://stacks.iop.org/0034-4885/75/i=1/a=016601

[15] S. L. Anna, "Droplets and bubbles in microfluidic devices," *Annual Review of Fluid Mechanics*, vol. 48, pp. 285–309, 2016.

[16] Z. Z. Chong, S. H. Tan, A. M. Gañán-Calvo, S. B. Tor, N. H. Loh, and N.-T. Nguyen, "Active droplet generation in microfluidics," *Lab Chip*, vol. 16, pp. 35–58, 2016. [Online]. Available: http://dx.doi.org/10.1039/C5LC01012H

[17] A. Karbalaei, R. Kumar, and H. J. Cho, "Thermocapillarity in microfluidics — a review," *Micromachines*, vol. 7, pp. 1–41, 2016. [Online]. Available: http://www.mdpi.com/2072-666X/7/1/13

[18] T. Toyota, N. Maru, M. M. Hanczyc, T. Ikegami, and T. Sugawara, "Self-propelled oil droplets consuming "fuel" surfactant," *Journal of American Chemical Society*, vol. 131, no. 14, pp. 5012–5013, 2009.

[19] I. Lagzi, S. Soh, P. J. Wesson, K. P. Browne, and B. A. Grzybowski, "Maze solving by chemotactic droplets," *Journal of American Chemical Society*, vol. 132, no. 4, pp. 1198–1199, 2010.

[20] S. Yabunaka, T. Ohta, and N. Yoshinaga, "Self-propelled motion of a fluid droplet under chemical reaction," *The Journal of Chemical Physics*, vol. 136, p. 074904, 2012. [Online]. Available: http://dx.doi.org/10.1063/1.3685805

[21] J. **C**ejková, M. Novák, F. **St**epánek, and M. M. Hanczyc, "Dynamics of chemotactic droplets in salt concentration gradients," *Langmuir*, vol. 30, pp. 11 937–11 944, 2014.

[22] H. R. Joshi, "Optimal control problems in pde and ode systems," Ph.D. dissertation, University of Tennessee-Knoxville, 2002.

[23] E. Casas, "Optimal control of pde theory and numerical analysis," pp. 1–136, August-September 2006, cIMPA School on Optimization and Control. [Online]. Available: https://cel.archives-ouvertes.fr/cel-00392187

[24] K. Kunisch and L. Wang, "Time optimal control of the heat equation with pointwise control constraints," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 19, no. 2, pp. 460–485, 2013.

[25] R. Herzog and K. Kunisch, "Algorithms for PDE-constrained optimization," *GAMM-Mitteilungen*, vol. 33, no. 2, pp. 163–176, January 2014.

[26] K. Chudej, H. J. Pesch, M. Wächter, G. Sachs, and F. Le Bras, *Instationary Heat-Constrained Trajectory Optimization of a Hypersonic Space Vehicle by ODE–PDE-Constrained Optimal Control*. New York, NY: Springer New York, 2009, pp. 127–144. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-95857-6_8

[27] M. Gerdts and S.-J. Kimmerle, "Numerical optimal control of a coupled ode–pde model of a truck with a fluid basin," *Dynamical Systems, Differential Equations and Applications AIMS Proceedings*, pp. 515–524, 2015.

[28] S.-J. Kimmerle, M. Gerdts, and R. Herzog, "Optimal control of an elastic crane-trolley-load system — a case study for optimal control of coupled ODE-PDE systems," Universit at der Bundeswehr M unchen, Tech. Rep., June 2015.

[29] M. H. Korayem, M. Haghpanahi, H. N. Rahimi, and A. Nikoobin, "Finite element method and optimal control theory for path planning of elastic manipulators," in *New Advances in Intelligent Decision Technologies: Results of the First KES International Symposium IDT 2009*, K. Nakamatsu, G. Phillips-Wren, L. C. Jain, and R. J. Howlett, Eds. Springer, 2009, pp. 117–126.

[30] P. Boscariol, "Dynamics and control of flexible-links mechanisms," Ph.D. dissertation, Università degli Studi di Udine, 2011.

[31] J. Petereit and T. Bernard, "Real–time nonlinear model predictive control of a glass forming process using a finite element model," in *Proc. IFIP Conference on System Modeling and Optimization*, ser. Lecture Notes in Computer Science. Springer, 2011, pp. 266–275.

[32] P. Boscariol and A. Gasparetto, "Model-based trajectory planning for flexible-link mechanisms with bounded jerk," *Robotics and Computer-Integrated Manufacturing*, vol. 29, pp. 90–99, 2013.

[33] R. Dubay, M. Hassan, C. Li, and M. Charest, "Finite element based model predictive control for active vibration suppression of a one-link flexible manipulator," *ISA Transactions*, vol. 53, no. 5, pp. 1609–1619, 2014.

[34] P. Bochev and D. Ridzal, "Finite element solution of optimal control problems arising in semiconductor modeling," pp. 1–8. [Online]. Available: http://www.cs.sandia.gov/~pbboche/papers_pdf/2007LSSC_Control.pdf

[35] E. Balsa-Canto, A. A. Alonso, L. T. Antelo, A. Arias-Méndez, E. López-Quiroga, D. Rivas, and C. Vilas, "Model identification and on-line optimal control of food processes," in *Coupled Problems 2013 Conference*, S. Idelsohn, N. Papadrakakis, and B. Schrefler, Eds., 2013, pp. 1–12, v International Conference on Computational Methods for Coupled Problems in Science and Engineering.

[36] M. Grundelius, "Methods for control of liquid slosh," Ph.D. dissertation, Lund Institute of Technology, 2001.

[37] Y. Tassa, T. Erez, and B. Smart, "Receding horizon differential dynamic programming," in *Advances in Neural Information Processing Systems 2008*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds., 2008, pp. 1465–1472. [Online]. Available: http://papers.nips.cc/paper/3297-receding-horizon-differential-dynamic-programming.pdf

[38] L. Van den Broeck, M. Diehl, and J. Swevers, "Time optimal mpc for mechatronic applications," in *Proceedings of the IEEE Conference on Decision and Control*, December 2009, pp. 8040–8045.

[39] R. Quirynen, M. Vukov, and M. Diehl, *Multiple Shooting in a Microsecond*. Springer International Publishing, 2015, pp. 183–201.

[40] G. Williams, A. Aldrich, and E. Theodorou, "Model predictive path integral control using covariance variable importance sampling," *CoRR*, vol. abs/1509.01149, 2015. [Online]. Available: http://arxiv.org/abs/1509.01149

[41] CHEM-IT European Commission FP7 FET Proactive Intiative: Bio-chemistry based Informatoin Technology. [Online]. Available: http://cordis.europa.eu/fp7/ict/fet-proactive/chemit_en.html

[42] MATCHIT 249032 European Commission FP7-ICT Project: Matrix for Chemical IT. [Online]. Available: http://fp7-matchit.eu/

[43] EVOBLISS 611640 European Commission FP7-ICT Project: Technological Evolution of Synergy Between Physicochemical and Living Systems. [Online]. Available: https://blogit.itu.dk/evoblissproject/

[44] J. M. P. Gutiérrez, "Automatic liquid handling for artificial life research," pp. 1–155, June 2012, MSc Thesis, Supervisors: Kasper Støy and David Brandt.

[45] J. M. P. Gutierrez, T. Hinkley, J. W. Taylor, K. Yanev, and L. Cronin, "Evolution of oil droplets in a chemorobotic platform," *Nature Communications*, vol. 5, no. 5571, 2014.

[46] National science foundation eccs, 1308025, droplet ratchets: Low-cost parallel microfluidics. [Online]. Available: http://grantome.com/grant/NSF/ECCS-1308025

[47] National science foundation eccs, 1102280, droplet thermotaxis: A new platform technology for droplet-based microfluidic systems. [Online]. Available: http://grantome.com/grant/NSF/ECCS-1102280

[48] A. Davanlou, "Thermally induced motion, collision and mixing of levitated droplets," Ph.D. dissertation, University of Central Florida, 2015.

[49] Z. Jiao and N.-T. Nguyen, "Manipulation of droplet in a planar channel by periodic thermocapillary actuation," *Journal of Micromechanics and Microengineering*, vol. 18, no. 045027, pp. 1–9, 2008.

[50] A. L. Yarin, W. Liu, and D. Reneker, "Motion of droplets along thin fibers with temperature gradient," *Journal of Applied Physics*, vol. 91, no. 7, 2002.

[51] C. N. Baroud, J.-P. Delville, F. Gallaire, and R. Wunenburger, "Thermocapillary valve for droplet production and sorting," *Physical Review E*, vol. 75, p. 046302, April 2007. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.75.046302

[52] E. Vela, H. Moustapha, and S. Régnier, "Laser-induced thermocapillary convection for mesoscale manipulation," *International Journal of Optomechatronics*, vol. 3, no. 4, pp. 289–302, 2009.

[53] E. Verneuil, M. L. Cordero, F. Gallaire, and C. N. Baroud, "Laser-induced force on a microfluidic drop: Origin and magnitude," *Langmuir*, vol. 25, no. 9, pp. 5127–5134, 2009. [Online]. Available: http://dx.doi.org/10.1021/la8041605

[54] D. E. Lucchetta, F. Simoni, L. Nucara, and R. Castagna, "Controlled-motion of floating macro-objects induced by light," *AIP Advances*, vol. 5, 2015.

[55] F. Takabatake, K. Yoshikawa, and M. Ichikawa, "Communication : Mode bifurcation of droplet motion under stationary laser irradiation," *The Journal of Chemical Physics*, vol. 141, no. 051103, pp. 10–14, 2014.

[56] U. Baur, P. Benner, and L. Feng, "Model order reduction for linear and nonlinear systems: a system-theoretic perspective," Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Tech. Rep., March 2014, mPIMD/14-07.

[57] M. M. Hanczyc, J. M. Parrilla, A. Nicholson, K. Yanev, and K. Stoy, "Creating and maintaining chemical artificial life by robotic symbiosis," *Artificial life*, vol. 21, no. 1, pp. 47–54, 2015.

[58] A. Javier, *Laser and Gaussian Beam Propagation and Transformation.* Marcel Dekker, Inc., 2003, pp. 999–1013.

[59] A. I. Fedosov, "Thermocapillary motion," *Zhurnal Fizicheskoi Khimii*, vol. 30, no. 2, pp. 366–373, 1956. [Online]. Available: http://arxiv.org/abs/1303.0243

[60] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, 7th ed. Oxford: Butterworth-Heinemann, 2013. [Online]. Available: http://www.sciencedirect.com/science/book/9781856176330

[61] L. H. Dill and R. Balasubramaniam, "Unsteady thermocapillary migration of bubbles," pp. 1–13, January 1988, nASA Technical Memorandum 101338.

[62] N. Rashidnia and R. Balasubramaniam, "Thermocapillary migration of liquid droplets in a temperature gradient in a density matched system," *Experiments in Fluids*, vol. 11, no. 2-3, pp. 167–174, 1991.

[63] R. Sun, "A nonlinear thermocapillary migration of droplets due to dependence of physical properties on temperature," *Journal of Hydrodynamics, Ser. B*, vol. 18, no. 4, pp. 469–474, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1001605806601224

[64] R. Eötvös, "Ueber den zusammenhang der oberflächenspannung der flüssigkeiten mit ihrem molecularvolumen," *Annalen der Physik*, vol. 263, no. 3, pp. 448–459, 1886.

[65] E. F. Greco and R. O. Grigoriev, "Thermocapillary migration of interfacial droplets," *Physics of Fluids*, vol. 21, 2009. [Online]. Available: http://dx.doi.org/10.1063/1.3112777

[66] E. L. Haseltine and J. B. Rawlings, "A critical evaluation of extended kalman filter and moving horizon estimation," Texas-Wisconsin Modeling and Control Consortium, Tech. Rep., March 2003.

[67] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear mpc and moving horizon estimation," *International Workshop on Assessment and Future Directions of NMPC*, pp. 1–24, September 2008.

[68] Makeblock. [Online]. Available: http://www.makeblock.com/

[69] T. P. Petrova, "Revised release on surface tension of ordinary water substance," The International Association for the Properties of Water and Steam, Moscow, Russia, Tech. Rep., June 2014, IAPWS R1-76(2014. [Online]. Available: http://www.iapws.org

[70] M. Witczak, *Unknown Input Observers and Filters*. Springer International Publishing, 2014, pp. 19–56.

[71] M. Hoshiya and A. Sutoh, "Kalman filter-finite element method in identification," *Journal of Engineering Mechanics*, vol. 119, no. 2, pp. 197–210, February 1993.

[72] H. Wakita, "Estimation of water flow at watarase retardation pond using kalman filter finite element method," Chuo University, Kawahara Lab., Tech. Rep., 2003.

[73] G. Battistelli, L. Chisci, N. Forti, G. Pelosi, and S. Selleri, "Distributed finite element Kalman filter," in *Proceedings of the European Control Conference (ECC)*, July 2015, pp. 3695–3700.

[74] F. Biral, E. Bertolazzi, P. Bosetti, A. De Marchi, and M. M. Hanczyc, "Optimal Control of a Laser Source to Generate a Minimum Time Trajectory of a Droplet in a Liquid Layer," in *Proc. IEEE Future Technologies Conference*, 2016, in Press.

[75] R. Lot and F. Biral, "A curvilinear abscissa approach for the lap time optimization of racing vehicles," in *Proc. 19th IFAC World Congress*, vol. 47, no. 3, Cape Town, South Africa, August 2014, pp. 7559–7565.

[76] F. Biral, E. Bertolazzi, and P. Bosetti, "Notes on numerical methods for solving optimal control problems," *IEEJ Journal of Industry Applications*, vol. 5, no. 2, pp. 154–166, 2016.

[77] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. Mishchenko, *The mathematical theory of optimal processes (International series of monographs in pure and aplied mathematics*. Interscience Publishers, 1962.

[78] J. D. Gibson, "A direct search approach to optimization for nonlinear model predictive control," *Optimal Control Applications and Methods*, vol. 36, pp. 139–157, 2015.

[79] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966. [Online]. Available: http://dx.doi.org/10.1080/00207176608921369

[80] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[81] G. Droge and M. Egerstedt, "Adaptive time horizon optimization in model predictive control," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011.

FTC 2016 - Future Technologies Conference 2016
6-7 December 2016 | San Francisco, United States

Optimal Control of a Laser Source to Generate a Minimum Time Trajectory of a Droplet in a Liquid Layer

Francesco Biral*, Enrico Bertolazzi*, Paolo Bosetti*, Alberto De Marchi* and Martin M. Hanczyc†
*Department of Industrial Engineering
University of Trento
Trento, Italy I-38123
Email: paolo.bosetti@unitn.it
†Centre for Integrative Biology
University of Trento
Trento, Italy I-38123
Email: martin.hanczyc@unitn.it

***Abstract*—This paper describes the use of optimal control theory applied to the motion of a low-power laser beam to precisely move a droplet in a liquid layer by means of the force on the liquid-liquid interface arising due to the surrounding thermal field (thermocapillary force and Marangoni effect). The form of heat field equations is approximated by the solution of a finite sum of laser pulses, whose analytical solution can be explicitly derived. In this way, the model of the system does not contain any partial differential equation (PDE). Optimization is performed by using the direct approach method and the *fmincon* solver provided by Matlab. The initial guess is found with a simple model predictive controller. Perspective applications of this technology include beam-controlled targeting of pharmaceuticals in organic tissues.**

Keywords*—Optimal control, model predictive control, droplet, thermocapillary force, targeting of pharmaceuticals

I. Introduction

The manipulation of liquid droplets suspended in another liquid can be achieved through thermocapillary motion through a nonuniform heat distribution in the system [1]. This nonuniformity results in thermocapillary stresses near the droplet interface with liquid flowing from the hot (lower surface tension) to cold side (higher surface tension) through Marangoni flows. In systems where viscosity dominates, the droplet will migrate. This resulting fluid flow dynamics allows for the movement of the droplet directionally through the system (e.g. for bubbles see [2]; for liquid droplets see [3]; for liquid droplets of spherical and aspherical shape see [4] and [5]).

The manipulation of droplets through heat gradient induced Marangoni flows have found practical applications in microfluidics including controlled mixing, confining, filtering, trapping and pumping of droplets [6]. The application of light energy to induce quick and controlled thermal gradients has also been explored [7], [8], [9]. By using optothermal capillary control, multidroplet manipulation and fusion has been demonstrated [10] as well as high speed droplet sorting [11].

The Authors have been exploring the dynamical motion of liquid droplets in a second immiscible liquid phase [12]. These droplets either have on board chemical potential which then induces and fuels a Marangoni flow system for autonomous motion [13], or the droplets are more passive and only respond to local chemical gradients in the system. Even in the simpler passive system the dynamical fluid motion of droplets allows for the completion of simple tasks such as solving mazes [14].

Interfacing dynamic droplets with electromechanical control can allow for the precise manipulation of droplets [10], [11], the sustenance of non-equilibrium states [15], and the exploration of the dynamical phase spaces [16].

In the present work, the Authors develop the use of optimal control theory applied to the motion of a low-power laser beam to precisely move a droplet through thermocapillary action. This methodology has application in the broader context of fluid mixing, tuning of interfacial properties and combinatorial processes including reactive combinatorial chemistry when droplets are precisely positioned to fuse. This will allow the control of sensitive chemical reactions on demand under controlled laboratory conditions as applied to biochemical analyses and material synthesis.

Currently this technology will add a fundamental control layer to our current EU project EVOBLISS that uses a robotic platform for the manipulation of chemical droplets [17]. In order to provide effective functionality for the project the optimal control mechanism will be implemented in real time to interface with the dynamics of the chemical droplets [15]. This is a novel substantiation of such an approach and the real time component is critical for precise manipulation of far from equilibrium chemical systems. This type of platform and control mechanism is distinct from often used microfluidics where microsized droplets are produced and manipulated almost exclusively through the imposed architectures of the microfluidic device [18], [19]. Instead here it is of interest to take advantage of targeted control mechanisms to temporarily interface with the dynamic droplets but then once the control is achieved release the droplets to allow them perform tasks more autonomously. It is hoped that in the future the precise positional control of reactive chemistries can be effected in complex environments such as organ tissues to deliver and

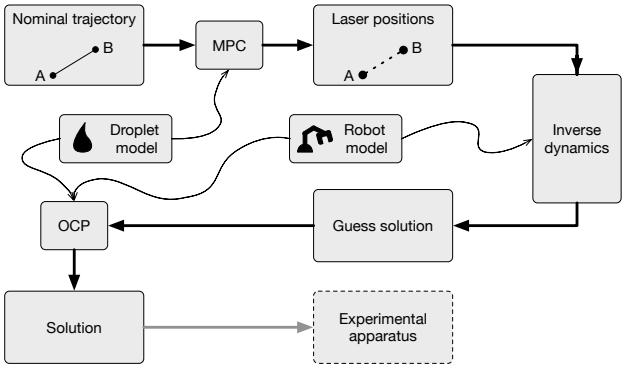


Fig. 1. Schematics of control method

perhaps trigger the synthesis and release of therapeutics on site.

II. Modeling

The work here presented is based on the idea of moving a droplet in its fluid environment by "pushing" it with the thermal gradient that results from heating the surrounding liquid with a laser beam. It is as letting the droplet surfing on the wave produced by a localized heat source.

In order to control the motion of the droplet by moving the laser spot about its proximity, the *trajectory* (i.e. the path and its time-history) of the laser spot itself has to be calculated according to the dynamics of the droplet and of the thermal field itself, so that the resulting laser motion is neither too fast—so that the droplet would loose tracking—nor too slow—which would result in the laser beam hitting the droplet itself.

Consequently, it is needed a set of analytical models (perhaps approximated) of the droplet and of the system actuating the laser position—which will be referred to as the *robot* in the followings.

These models are then used for calculating the optimum time-sequence of *controls* that ensure the robot moves the laser at the optimal speed (neither too fast nor too slow) and along the desired trajectory in its liquid medium.

The schematics in Fig. 1 summarizes the approach: the whole process starts from the initial definition of the desired, nominal trajectory. With the approach known as *Model Predictive Control* (MPC), the model of the droplet dynamics is then used for calculating the time-sequence of laser spot positions that would drive the droplet from its initial position to the target one. This sequence of position is then converted in a sequence of robot *controls* (or inputs to the robot motors) thanks to the robot model and to its inverse dynamics. The result of this step is the *guess solution*, i.e. a tentative, approximate solution that still does not ensure an optimal performance, and which is eventually used for solving the *Optimal Control Problem* (OCP). The latter, finally, produces an optimal time sequence of controls that allows to drive the droplet from A to B maximizing the performance and yet complying with a given set of constraints.

The following sections are detailedly describing the models of the dynamics for droplet and robot, and the numerical approaches taken for solving the two-step optimization approach (MPC + OCP).

A. Model of Laser Heating

For a source of intensity I_0, the transmitted intensity I of an electromagnetic wave penetrating a material is given by Beer's law

$$I(z) = I_0 e^{-\beta(\lambda) z} \tag{1}$$

where $\beta(\lambda)$ is the absorption coefficient that depends on laser wavelength λ and z is the path length. Let us focus on an absorbing layer of thickness Δz along the beam propagation direction. For a weakly absorbing medium, i.e. $\beta(\lambda)\Delta z \ll 1$, the intensity $I(z)$ can be approximated as a first-order term

$$I(z) \approx I_0 \left[1 - \beta(\lambda) z\right] \tag{2}$$

Therefore the intensity reduction per unit thickness is $\beta(\lambda) I_0$. Assuming that all the light energy absorbed by the medium is converted instantly to heat, the intensity reduction per unit thickness is the thermal energy deposited in the medium per unit volume per unit time

$$\dot{q} = \beta(\lambda) I_0 \tag{3}$$

Let us define temperature field w, laser position $\mathbf{x}_L$ and power p_L, heat sink temperature $\bar{w}$ and the spatial domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$, time-space domain $\tilde{\Omega} = (t_i, t_f) \times \Omega$ with spatial boundary $\partial\tilde{\Omega} = (t_i, t_f) \times \tilde{\Omega}$. Transient thermal dynamics is governed by the unsteady heat equation assuming constant liquid properties (specific mass ρ, specific heat c_p, thermal conductivity k, thermal diffusivity $\alpha = k/\rho c_p$) and absence of any mass transport. For $(t, \mathbf{x}) \in \tilde{\Omega}$ it reads

$$\rho c_p \frac{\partial w}{\partial t}(t, \mathbf{x}) = k \nabla^2 w(t, \mathbf{x}) + \dot{q}(t, \mathbf{x}) \tag{4}$$

For the sake of simplicity, constant homogeneous Dirichlet boundary conditions, i.e. $\bar{w}(t, \mathbf{x}) = 0 \ \forall (t, \mathbf{x}) \in \partial\tilde{\Omega}$, are selected. Moreover, laser-induced incident intensity is assumed to be Gaussian with characteristic radius r_L and with fixed wavelength, so that absorption coefficient $\beta = \beta(\lambda)$ is constant.

$$\dot{q}(t, \mathbf{x}) = \frac{2\beta}{\pi r_L^2} \, p_L \, e^{-2\frac{\|\mathbf{x} - \mathbf{x}_L\|^2}{r_L^2}} \tag{5}$$

So the governing equation expressed in polar coordinates (centred in laser pulse position) reads

$$\rho c_p \frac{\partial w}{\partial t}(t, \mathbf{x}) = k \nabla^2 w(t, \mathbf{x}) + \frac{2\beta p_L}{\pi r_L^2} \, e^{-2\frac{r(\mathbf{x})^2}{r_L^2}} \, \delta(t - t_i) \tag{6}$$

Let us recall the solution $\tilde{w}$ for an instantaneous point-wise heat release (energy $p_L \Delta t$ in $r = 0$ at time $t = 0$) and build by similarity a function θ with free parameters a, b and c

$$\begin{aligned} \tilde{w} &= \bar{w} + \frac{\beta p_L \Delta t}{\pi \rho c_p} \frac{1}{4\alpha t} e^{-r^2/(4\alpha t)} \\ \theta &= \bar{w} + \frac{a}{ct + 1} e^{-br^2/(ct+1)} \end{aligned} \tag{7}$$

Considering the function θ, it shall be assessed if it could be a solution of the heat equation above, and in that case its coefficients will be matched. Substituting θ and its derivatives

into the heat equation, after some calculations, one gets two conditions on coefficients:

$$c = 4\alpha b \qquad b = \frac{2}{r_L^2}, \tag{8}$$

where $\alpha = k/(\rho c_p)$ is the liquid thermal diffusivity. Then, by the energy conservation law, one can also write:

$$\begin{aligned} \beta p_L \Delta t &= \int_\Omega \rho c_p \left(\theta - \bar{w}\right) d\Omega \\ &= \int_0^{2\pi} \int_0^{\infty} \rho c_p \frac{a}{ct+1} e^{-br^2/(ct+1)} r dr d\theta \\ &= \pi \rho c_p \frac{a}{b} \\ &\Downarrow \\ a &= \frac{\beta p_L \Delta t}{\pi \rho c_p} b \end{aligned} \tag{9}$$

Thus, by solving for coefficients a, b and c and substituting into function θ, one find the analytical expression of temperature field due to an instantaneous Gaussian laser pulse. Notice that for every finite r_L the solution satisfies:

$$\begin{aligned} \lim_{r\to\infty} \theta &= \bar{w} \\ \lim_{t\to\infty} \theta &= \bar{w} \end{aligned} \tag{10}$$

For $t > 0$, it results:

$$w(t, \mathbf{x}) = \bar{w} + \frac{2}{\pi} \frac{\beta p_L \Delta t}{\rho c_p} \frac{1}{8\alpha t + r_L^2} e^{-2\frac{r(\mathbf{x})^2}{8\alpha t + r_L^2}} \tag{11}$$

where $r(\mathbf{x})$ is the distance between $\mathbf{x}$ and heat release position $\mathbf{x}_L$.

Then, the exact thermal field generated by a laser beam trajectory $\mathbf{x}_L(t)$ and laser power profile $p_L(t)$ can be computed by the convolution integral (thanks to linearity), for $t > 0$. Actually it turns out that this integral is quite involved, so it is preferable to build an equispaced time grid and approximate the convolution integral as the finite sum of instantaneous laser pulses.

$$\begin{aligned} w(t, \mathbf{x}) &= \bar{w} + \frac{2}{\pi} \frac{\beta}{\rho c_p} \int_0^t \frac{p_L(\tau)}{8\alpha(t-\tau) + r_L^2} e^{-2\frac{\|\mathbf{x} - \mathbf{x}_L(\tau)\|^2}{8\alpha(t-\tau) + r_L^2}} d\tau \\ &\Downarrow \\ w(t_k, \mathbf{x}) &\approx \bar{w} + \frac{2}{\pi} \frac{\beta \Delta t}{\rho c_p} \sum_{j=0}^{k} \frac{p_{L,j}}{8\alpha(k-j)\Delta t + r_L^2} e^{-2\frac{\|\mathbf{x} - \mathbf{x}_{L,j}\|^2}{8\alpha(k-j)\Delta t + r_L^2}} \end{aligned} \tag{12}$$

One can than compute explicitly the thermal gradient, considering a cylindrical system of coordinates for each laser pulse and projecting then in the absolute Cartesian system:

$$\begin{aligned} &\nabla w(t_k, \mathbf{x}) = \\ &= -\frac{8}{\pi} \frac{\beta \Delta t}{\rho c_p} \sum_{j=0}^{k} \frac{p_{L,j}(\mathbf{x} - \mathbf{x}_{L,j})}{\left[8\alpha(k-j)\Delta t + r_L^2\right]^2} e^{-2\frac{\|\mathbf{x} - \mathbf{x}_{L,j}\|^2}{8\alpha(k-j)\Delta t + r_L^2}} \end{aligned} \tag{13}$$

The approximation of temperature field using a finite sum of pulses depends on the time step Δt, that is the time interval between two consecutive laser pulses. In order to validate this approach and to estimate an upper bound for the time step, it is useful to compare the result with an analytical solution. Temperature profile induced by a continuous—not pulsed—point heat source moving at constant speed is known in literature. Approximations of this analytical solution with finite sum of pulses are shown in Fig. 2 and Fig. 3, considering two different time steps. Moreover, temperature field approximation in case of Gaussian laser beam is reported. Parameters are characteristic radius $r_L = 0.2\,\mathrm{mm}$, constant speed $10\,\mathrm{mm/s}$, power laser and liquid properties from I. It is evident that analytical and *finite-sum* profiles converge as time step diminishes. As expected, in the neighbourhood of laser spot, predicted temperature due to point or Gaussian source is quite different, being in the latter case much smoother. Thanks to this, the sensitivity of temperature approximation in the case of Gaussian source is smaller.

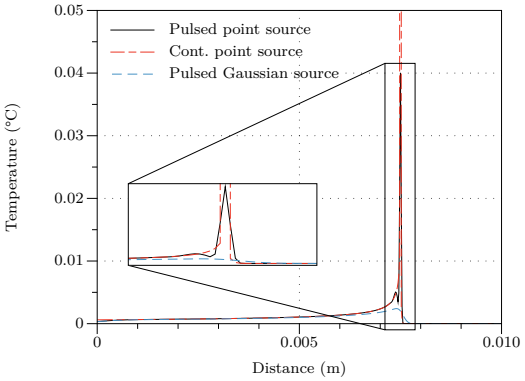


Fig. 2. Temperature field induced by laser heating: finite sum approximation with time step $\Delta t = 10\,\mathrm{ms}$ (spatial resolution $20\,\mu\mathrm{m}$)

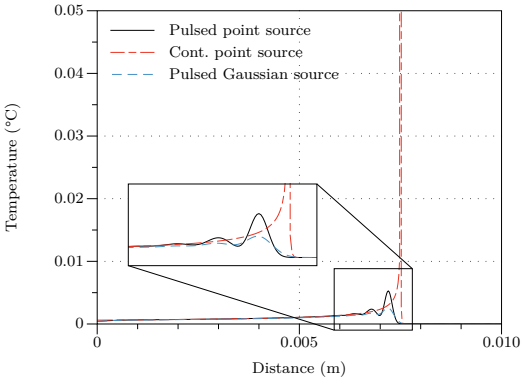


Fig. 3. Temperature field induced by laser heating: finite sum approximation with time step $\Delta t = 40\,\mathrm{ms}$ (spatial resolution $20\,\mu\mathrm{m}$)

B. Model of Thermocapillary Motion

1) Steady-state Migration Velocity: From classical works [1], [3], the thermocapillary migration velocity of a droplet on an open flat container with thin liquid layer is

$$\mathbf{u} = -\frac{2a}{(2 + \frac{\alpha\prime}{\alpha})(2\mu + 3\mu\prime)} \sigma_T \nabla w \tag{14}$$

where μ is the dynamic viscosity of the liquid and α its thermal diffusivity, a is the droplet radius, $\frac{\partial \sigma}{\partial w} = \sigma_T$ the thermal interfacial tension coefficient, w the temperature at droplet position (neglecting the presence of droplet itself). Terms with prime relate to droplet properties, without to liquid properties.

2) Unsteady Thermocapillary Motion: Detailed dynamics can be captured solving Navier-Stokes equations. This problem is widely covered in literature in the approximation of small Reynolds and Marangoni numbers [2]. Actually, estimating Reynolds number with $\rho \approx 1000\,\mathrm{kg/m^3}$, $v \approx 2\,\mathrm{mm/s} \div 5\,\mathrm{mm/s}$, $l \approx 2\,\mathrm{mm} \div 4\,\mathrm{mm}$ and $\mu \approx 0.8\,\mathrm{mPa}$, one gets $Re = \rho v l/\mu \approx 5 \div 25$. Thus, flow can be considered laminar but transient phase is not negligible. Simple models that fit this dynamics with experimental evidence have been found [9].

Lets now apply Newton's approach assuming the droplet to be a point-mass. Considering linear thermocapillary force and quadratic drag force, the droplet equation of motion reads as:

$$\begin{aligned} \dot{\mathbf{x}}_D(t) &= \mathbf{v}_D(t) \\ m\dot{\mathbf{v}}_D(t) &= \mathbf{F}_\sigma(t) + \mathbf{F}_{drag}(t) \\ &= c_1 a^2 \sigma_T \nabla w(t, x_D) - c_2 a^2 \|\mathbf{v}_D(t)\| \mathbf{v}_D(t) \end{aligned} \tag{15}$$

where x_D is droplet position, v_D droplet velocity, m mass, a radius, $\sigma_T = \frac{\partial \sigma}{\partial w}$ liquid-droplet surface tension coefficient, w temperature, c_1 and c_2 scalar parameters. Constant $c_2 = \rho_l C_{drag}/2 > 0$ collects liquid specific mass ρ_l and drag coefficient C_{drag}. All liquid and droplet properties are considered constant with respect to temperature and time.

Constant c_1 may be estimated considering the steady-state condition $\dot{\mathbf{v}}_D = \mathbf{0}$, i.e. when subject to a constant thermal gradient $\widetilde{\nabla w}$ the droplet reaches the steady-state velocity $\mathbf{v}_D^\infty$. In this condition it becomes:

$$\begin{aligned} 0 &= c_1 a^2 \sigma_T \widetilde{\nabla w} - c_2 a^2 \|\mathbf{v}_D^\infty\| \mathbf{v}_D^\infty \\ &\Downarrow \\ c_1 &= \frac{c_2}{\sigma_T} \|\mathbf{v}_D^\infty\| \mathbf{v}_D^\infty \widetilde{\nabla w}^{-1} \end{aligned} \tag{16}$$

Experimentally it was possible to find $\|\mathbf{v}_D^\infty\| \approx 5$ mm/s and then numerically to estimate $\widetilde{\nabla w} \approx -15$ K/m. Other parameters of the experimental setup are reported in Tab. I. Finally, as results from (16), the estimation of thermocapillary force constant is $c_1 \approx 2$.

Lets now consider now the definition of Marangoni number $Ma = -\sigma_T L \Delta w/(\mu\alpha)$—$L$ and Δw characteristic length and temperature difference, μ dynamic viscosity—and try to estimate the surface tension force as $F_\sigma = Ma\mu\alpha = -\sigma_T L \Delta w$. Assuming $\Delta w = -a\nabla w$ and $L = a$ as characteristic quantities, one finds exactly $c_1 = 1$, meaning that the order-of-magnitude of the previous estimation may be correct.

C. Model of Robotic Platform

The robotic platform involved in this project is essentially a 3D printer, with a modular head for syringes, laser and other tools. The kinematics is planar and cartesian and two stepper motors move one axle each. Robot state is described by laser position $\mathbf{x}_L$ and velocity $\dot{\mathbf{x}}_L$ and control inputs are electric signals $\mathbf{I}_L$. Lets consider a generic second-order system to represent the platform, that is:

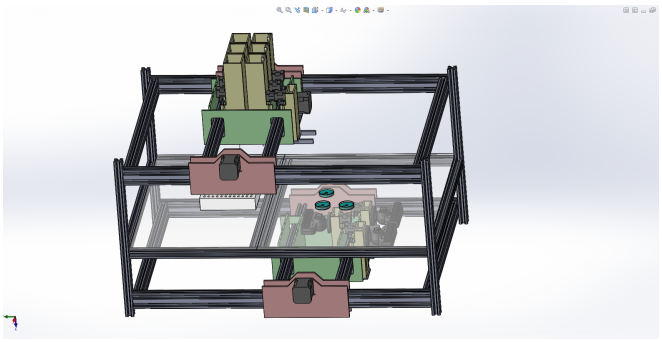


Fig. 4. Robotic platform developed for EVOBLISS project [17]

$$\begin{aligned} \dot{\mathbf{x}}_L &= \mathbf{v}_L \\ M_L(\mathbf{x}_L)\dot{\mathbf{v}}_L &= B_L(\mathbf{x}_L)\mathbf{I}_L - C_L(\mathbf{x}_L, \mathbf{v}_L)\mathbf{v}_L \end{aligned} \tag{17}$$

where it is defined mass matrix M_L, input matrix B_L, damping matrix C_L—not diagonal in general. This model is quite general and comprises many different types of systems. Actually, it reasonably represents stepper motor dynamics under non-slipping conditions. Then, having the platform two independent axles, matrices M_L, B_L and C_L become diagonal. Moreover, the two axes are considered to be equal and independent on the state, so one can replace matrices with three scalar values—m_L, b_L and c_L. Finally, the model reads as:

$$\begin{aligned} \dot{\mathbf{x}}_L &= \mathbf{v}_L \\ m_L\dot{\mathbf{v}}_L &= b_L\mathbf{I}_L - c_L\mathbf{v}_L \end{aligned} \tag{18}$$

III. Optimal Control Problem

A. Formulation

Lets now collect states in vector $\mathbf{z} = [\mathbf{x}_D, \mathbf{v}_D, \mathbf{x}_L, \mathbf{v}_L]^T$ and controls $\mathbf{u} = \mathbf{I}_L$ and define the cost functional J for the time-optimal control problem, with cost on control and laser pointer velocity.

$$\begin{aligned} J &= \int_{t_i}^{t_f} l(t, \mathbf{z}, \mathbf{u}) dt \\ l(t, \mathbf{z}, \mathbf{u}) &= 1 + \frac{w_u}{2}\|\mathbf{u}\|^2 + \frac{w_v}{2}\|\mathbf{v}_L\|^2 \end{aligned} \tag{19}$$

Then, in order to have a fixed final time problem, one applies the linear transformation $t = t_i + (t_f - t_i)\tau$, so that the independent variable becomes $\tau \in [0, 1]$. Doing this, final time t_f becomes a parameter to be optimized and the model can be

written as follows:

$$\begin{aligned}
\frac{d\mathbf{x}_D}{d\tau}(\tau) &= (t_f - t_i)\mathbf{v}_D(\tau) \\
m_D\frac{d\mathbf{v}_D}{d\tau}(\tau) &= (t_f - t_i)\left[b_D\nabla w(\tau, \mathbf{x}_D(\tau)) - c_D\mathbf{v}_D(\tau)\right] \\
\frac{d\mathbf{x}_L}{d\tau}(\tau) &= (t_f - t_i)\mathbf{v}_L(\tau) \\
m_L\frac{d\mathbf{v}_L}{d\tau}(\tau) &= (t_f - t_i)\left[b_L\mathbf{u}(\tau) - c_L\mathbf{v}_L(\tau)\right] \\
w(\tau, \mathbf{x}) &= \bar{w} + \frac{2\beta}{\pi\rho c_p}\sum_j \frac{(t_f - t_i)p_{L,j}\delta\tau}{A_j(\tau)} e^{-\frac{2\|\mathbf{x}-\mathbf{x}_{L,j}\|^2}{A_j(\tau)}} \\
A_j(\tau) &= 8\alpha(\tau - \tau_j)(t_f - t_i) + r_L^2
\end{aligned} \tag{20}$$

subject to initial and final conditions:

$$\begin{aligned}
\mathbf{x}_D(0) &= \mathbf{x}_D^i & \mathbf{v}_D(0) &= \mathbf{v}_D^i \\
\mathbf{x}_D(1) &= \mathbf{x}_D^f & \mathbf{v}_D(1) &= \mathbf{v}_D^f \\
\mathbf{x}_L(0) &= \mathbf{x}_L^i & \mathbf{v}_L(0) &= \mathbf{v}_L^i \\
\mathbf{x}_L(1) &= \mathbf{x}_L^f & \mathbf{v}_L(1) &= \mathbf{v}_L^f
\end{aligned} \tag{21}$$

In compact form it is:

$$\begin{aligned}
\frac{d\mathbf{z}}{d\tau}(\tau) &= \boldsymbol{\varphi}(\tau, t_f, \mathbf{z}, \mathbf{u}(\tau)) \\
\mathbf{z}(0) &= \mathbf{z}^i \\
\mathbf{z}(1) &= \mathbf{z}^f
\end{aligned} \tag{22}$$

B. Discretisation and Gradient Evaluation

Discretisation of cost function and dynamical constraints makes the optimal control problem a finite dimensional optimisation problem. Building an equispaced time grid, $\tau_k = k\Delta\tau$, and by using explicit forward Euler scheme one can approximate the solution of the ODE that describes the system dynamics, $\mathbf{z}(\tau_k) \approx \mathbf{z}_k$.

$$\begin{aligned}
&\frac{d\mathbf{z}}{d\tau}(\tau) = \boldsymbol{\varphi}(\tau, t_f, \mathbf{z}, \mathbf{u}(\tau)) \\
&\Downarrow \\
&\frac{\mathbf{z}_{k+1} - \mathbf{z}_k}{\Delta\tau} = \boldsymbol{\varphi}(\tau_k, t_f, \mathbf{z}_0, \ldots, \mathbf{z}_k, \mathbf{u}_k) \\
&\Downarrow \\
&\mathbf{z}_{k+1} = \mathbf{z}_k + \boldsymbol{\varphi}(\tau_k, t_f, \mathbf{z}_0, \ldots, \mathbf{z}_k, \mathbf{u}_k)\Delta\tau
\end{aligned} \tag{23}$$

Defining the vector of variables $\mathbf{x}$, one can easily express cost function J, equality $\mathbf{g}_{eq}$ and inequality $\mathbf{g}$ constraints in the form requested by *fmincon* function provided by Matlab:

$$\mathbf{x} = [t_f, \mathbf{z}_0, \ldots, \mathbf{z}_N, \mathbf{u}_0, \ldots, \mathbf{u}_N]^T \tag{24}$$

Actually, it is possible to give an analytical expression for the derivatives of these functions, and it is useful to reduce the computational effort:

$$\begin{aligned}
&J[\mathbf{u}] = (t_f - t_i) + \int_{t_i}^{t_f}\left(\frac{w_u}{2}\|\mathbf{u}(t)\|^2 + \frac{w_v}{2}\|\mathbf{v}_L(t)\|^2\right)dt \\
&\Downarrow \\
&J(\mathbf{x}) = (t_f - t_i)\left[1 + \Delta\tau\sum_{k=0}^{N}\left(\frac{w_u}{2}\mathbf{u}_k^T\mathbf{u}_k + \frac{w_v}{2}\mathbf{v}_{L,k}^T\mathbf{v}_{L,k}\right)\right]
\end{aligned} \tag{25}$$

where $\Delta\tau = 1/N$. Dynamics and boundary conditions are considered as equality constraints:

$$\begin{aligned}
&\mathbf{g}_{eq}(\mathbf{x}) = \\
&= \begin{bmatrix}
\mathbf{z}_0 - \mathbf{z}^i \\
\mathbf{z}_N - \mathbf{z}^f \\
\mathbf{z}_1 - \mathbf{z}_0 - \boldsymbol{\varphi}(\tau_0, t_f, \mathbf{z}_0, \mathbf{u}_0)\Delta\tau \\
\vdots \\
\mathbf{z}_{k+1} - \mathbf{z}_k - \boldsymbol{\varphi}(\tau_k, t_f, \mathbf{z}_0, \ldots, \mathbf{z}_k, \mathbf{u}_k)\Delta\tau \\
\vdots \\
\mathbf{z}_N - \mathbf{z}_{N-1} - \boldsymbol{\varphi}(\tau_{N-1}, t_f, \mathbf{z}_0, \ldots, \mathbf{z}_{N-1}, \mathbf{u}_{N-1})\Delta\tau
\end{bmatrix}
\end{aligned} \tag{26}$$

Finally, cost function derivatives with respect to vector of variables $\frac{\partial J}{\partial \mathbf{x}}$ are:

$$\begin{aligned}
\frac{\partial J}{\partial t_f} &= 1 + \Delta\tau\sum_{k=0}^{N}\left(\frac{w_u}{2}\mathbf{u}_k^T\mathbf{u}_k + \frac{w_v}{2}\mathbf{v}_{L,k}^T\mathbf{v}_{L,k}\right) \\
\frac{\partial J}{\partial \mathbf{u}_j} &= (t_f - t_i)\Delta\tau w_u\mathbf{u}_j^T \\
\frac{\partial J}{\partial \mathbf{z}_j} &= (t_f - t_i)\Delta\tau w_v\mathbf{v}_{L,j}^T
\end{aligned} \tag{27}$$

Derivatives of constraints $\mathbf{g}_{eq}$ have been explicitly derived (not reported here).

C. Model Predictive Control

In this work MPC approach is used to generate a guess solution—i.e. a tentative, suboptimal solution—to initialise the optimisation process. MPC has the ability to anticipate future events—standard PID and LQR have not—and can take control actions accordingly (considering the evolution of a thermal field for instance). Suitable control actions are evaluated in the sense of minimizing a cost function. Thus, the model of the dynamical system is used for calculating step-by-step the optimal next laser spot position (neglecting robot dynamics), taking into account the evolution of the system in a finite time-horizon (MPC is also called *receding horizon control*).

Cost function considered here is the distance between the droplet and a point sliding along the reference trajectory (*sliding control*), computed at the time-horizon (*Mayer term*). In the case of a straight line, the choice of how much to slide is not critical—ensuring that it is large enough to avoid $\Delta x \approx \Delta x_D = v_D\Delta t$. However, this parameter may be time-varying and tuned to affect the high-level behaviour of the controller. For instance, in the case of zero final velocity of the droplet, saturation on the target position allows to estimate a better tentative solution with respect to a constant sliding (that is fine for free final velocity).

At each time step, given the actual state of the system, cost function has to be minimized and to this end *patternsearch* function provided by Matlab is used, because it manages constraints and it is a direct-search algorithm. Notice that cost function derivative with respect to the next laser position may be explicitly derived and used in a gradient-based optimizer, but it is quite involved and this approach is not exploited here.

Then, the calculated time-sequence of laser spot positions is used to reconstruct the control input using inverse dynamics

of the robot, Eq. (28). Finally, after checking the feasibility of control input, system evolution is computed again (if needed).

$$\begin{aligned} &\mathbf{x}_{L,k} \qquad k = 1, \dots, N \\ &\Downarrow \\ &\mathbf{v}_{L,k} = \frac{\mathbf{x}_{L,k+1} - \mathbf{x}_{L,k}}{\Delta t} \\ &\dot{\mathbf{v}}_{L,k} = \frac{\mathbf{x}_{L,k+1} - 2\mathbf{x}_{L,k} + \mathbf{x}_{L,k-1}}{\Delta t^2} \\ &\Downarrow \\ &\mathbf{I}_{L,k} = \frac{m_L}{b_L}\dot{\mathbf{v}}_{L,k} + \frac{c_L}{b_L}\mathbf{v}_{L,k} \end{aligned} \tag{28}$$

The result of this procedure is a feasible guess, i.e. a tentative solution that satisfies constraints—perhaps not final condition $\mathbf{z}(1) = \mathbf{z}^f$.

IV. Simulations

A. Set-up

The approach presented in previous Sections has been tested with two particular final conditions, viz. fixed and free droplet final velocity. Initially both the droplet and the robot are still and the latter is in the origin, final conditions of the robot are free and the reference path is a straight line. The time step used for generating the guess solution with MPC is $\Delta t = 50\,\mathrm{ms}$, but this only defines the refinement of the time grid parametrization. The actual time step depends on the final solution of OCP—on the optimal final time t_f in particular.

Key parameters in the MPC setup are the time horizon $t_h = 1\,\mathrm{s}$ and the sliding $\Delta s = 1$ (given a parametrization of the straight line with $s \in [0, 1]$). Tolerance on mesh size $TolMesh = 10\,\mu\mathrm{m}$ is provided to *patternsearch*, being a trade-off between accuracy and speed in guess solution generation.

In OCP formulation, parameters w_u and w_v define the relative importance of different terms in the cost functional J, Eq. (19). In simulations, unitary values of these weights are considered, i.e. $w_u = 1\,\mathrm{s/A^2}$ and $w_v = 1\,\mathrm{s^3/m^2}$. An *a posteriori* analysis demonstrates that influence of control and velocity cost is small compared to the final time cost, namely

$$N w_v \|\mathbf{v}_L\|^2 \approx N w_u \|\mathbf{u}\|^2 \ll t_f - t_i \tag{29}$$

Other parameters used in simulations are reported in Tab. I.

TABLE I. Simulation parameters

Robot	
$p_L = 0.5\,\mathrm{W}$	$r_L = 0.5\,\mathrm{mm}$
$m_L = 1\,\mathrm{kg}$	$b_L = 10\,\mathrm{N/A}$
$c_L = 10\,\mathrm{Ns/m}$	$\lvert i_L \rvert < 1\,\mathrm{A}$
Droplet	
$a = 2\,\mathrm{mm}$	$\rho = 1000\,\mathrm{kg/m^3}$
$\sigma_T = -0.18\,\mathrm{mN/Km}$	$C_{drag} = 0.4$
Liquid	
$\rho = 997\,\mathrm{kg/m^3}$	$c_p = 4186\,\mathrm{J/kgK}$
$k = 0.609\,\mathrm{W/Km}$	$\beta = 0.06\,\mathrm{m^{-1}}$
$\mu = 0.8\,\mathrm{mPas}$	

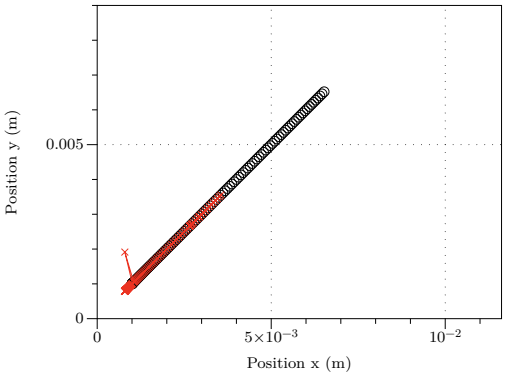


Fig. 5. Droplet and laser trajectories in the case of zero initial and free final velocities (see animation on https://vimeo.com/170754139)

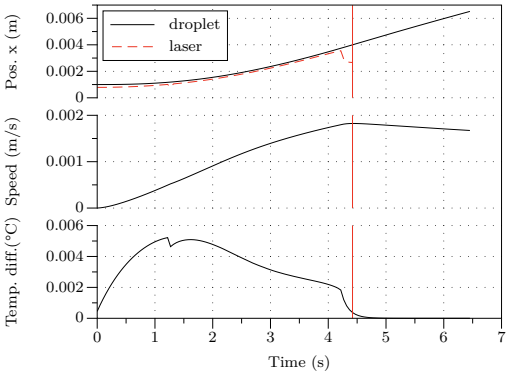


Fig. 6. Temperature, droplet speed, and x positions for laser and droplet in the case of zero initial and final velocities

B. Results and Discussion

Results are here discussed in the case of free and fixed final velocity of the droplet. In particular, results refer to the system evolution obtained considering only the control from the solution of the OCP. This step is done because the optimiser might generate a solution that violates dynamical constraints. Examples of typical laser and droplet trajectories obtained by simulations are shown in Fig. 5 and Fig. 7. It is possible to evaluate all the variables of interest, such as kinematic, dynamic and thermal quantities for instance. See Fig. 6 and Fig. 8 for these variables.

Let us discuss about the case of free final velocity. It is evident from Fig. 6 that there is an optimal distance between droplet and laser spot. One may analyse this problem considering the case of a continuous laser beam moving at constant speed $\tilde{u}$ along x axis and in quasi-stationary conditions ($\partial w/\partial t = 0$). By defining the spatial coordinate $\xi = x - \tilde{u}t$, the temperature gradient may be expressed as:

$$\nabla w = \nabla w(\xi, \tilde{u}) \tag{30}$$

Then, it can be noticed that in steady-state conditions droplet moves at the same speed $\tilde{u}$ and is subject to a constant thermal gradient $\widetilde{\nabla w} = \nabla w(\tilde{\xi}, \tilde{u})$. So there is a set of possible equilibria $(\tilde{\xi}, \tilde{u})$ defined by the coupling of droplet and thermal dynamics:

$$0 = c_1 \sigma_T \nabla w(\tilde{\xi}, \tilde{u}) - c_2 |\tilde{u}| \tilde{u} \tag{31}$$

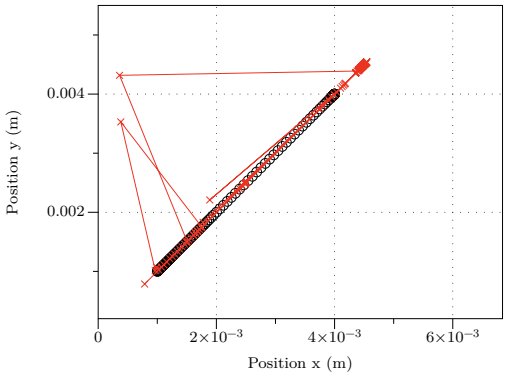


Fig. 7. Droplet and laser trajectories in the case of initial and final velocities both null (see animation on https://vimeo.com/170755486)

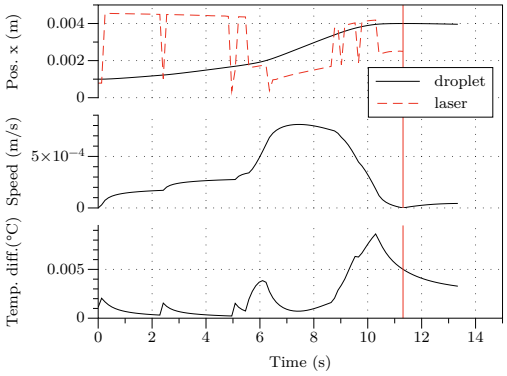


Fig. 8. Temperature, droplet speed, and x positions for laser and droplet in the case of initial and final velocities both null

Finally, one may formulate an optimisation problem aimed to maximize the steady-state velocity $\tilde{u}$ while satisfying the constraint expressed by Eq. (31). In the case of point source the analytical expression of $\nabla w(\xi, \tilde{u})$ is available, but the aforementioned optimisation problem does not have a simple closed-form solution. However, it is possible to evaluate numerically the optimal pair $(\tilde{\xi}_{max}, \tilde{u}_{max})$ (results omitted here for the sake of brevity).

It is worth to notice that steady state conditions are not reached in the simulated time slot; actually, it seems that the droplet is uniformly accelerated, because speed profile is almost linear. Temperature difference initially arises due to the laser heating, but then decreases as droplet (and robot) speed increases—because less energy per unit volume is released at higher speed. Finally, after stopping the laser heating, droplet is subject to drag and thermocapillary force induced by the residual unsteady temperature field, so without the dominant driving force the droplet slows down.

On the other hand, considering the case of zero final velocity, the optimal strategy is quite different. As reported in Fig. 8, the robot moves repeatedly from behind the droplet to the neighbourhood of the target position. In particular, the robot starts pushing the droplet, it moves close to the target, then it pushes again and finally adjusts the trajectory. So the optimal strategy may be to prepare a thermal gradient needed to stop the droplet while it is just started. It is expected that the timing of these steps would depend on the time constants and the characteristic length of the system. Temperature profile in this case has a maximum close to the final time, because of the initial heating to brake the droplet. Finally, one can notice the lack of symmetry in the laser spot trajectory, Fig. 7. Only few points are captured (due to large robot acceleration and large time step) but it is clear that the laser would shut down in those points, in order to not affect droplet dynamics.

As a final remark, it shall be highlighted the fact that reported results may correspond to local optima found as solution of the OCP.

The model above proposed for the thermal dynamics of the droplet/liquid system has been experimentally validated. In particular, one set of tests has been used for the identification of model parameters, and the resulting, identified model has been then validated by comparison with a different set of tests.

The validation setup is a Petri dish ($5\,\mathrm{mM}$, pH 11 decanoate solution, volume $9\,\mathrm{ml}$, $90\,\mathrm{mm}$ dish diameter) with a single droplet close to the dish center (lens-shaped droplet, volume $20\,\mu\mathrm{l}$, 1-decanol). The system temperature was constant and uniform within $0.05°\mathrm{C}$. An RGB camera has been used for capturing the test area in the Petri dish, using a calibrated image analysis system for measuring absolute and relative positions of droplet and laser beam image. The laser source was a $1.6\,\mathrm{W}$ power, $405\,\mathrm{nm}$ wavelength unit with a beam diameter of $5\,\mathrm{mm}$.

In the first set of tests, used for calibrating the model parameters, the laser was kept fixed pointing at position close, but not overlapping, to the droplet. The resulting bell-shaped thermal field was pushing the droplet away from the laser spot in radial direction. In the second set of tests, used for validation, the laser was slowly moved against the droplet thus guiding the droplet along a straight path.

Referring to the model in (13)–(15), the physical properties of the liquid medium are known, while the absorption coefficient β, the laser beam radius r_L, the thermocapillary sensitivity c_1, and the friction coefficient c_2 have to be identified. Being the model linear, the effects of β and b_D are undistinguishable from the droplet point-of-view, and only the βb_D product can be identified (see (13) and (15)). In order to take into consideration the actual droplet size (while in the model the droplet is a 1-D point), an additional parameter Δ is introduced to represent the distance between the laser spot and the droplet boundary.

Those parameters have been identified by optimization. In fact, given the system initial conditions, the set of known and guess parameters, and the laser trajectory, it is possible to calculate the evolution of the system according to its analytical model. Then, the actual droplet trajectory and the simulated one can be compared, and the distance between laser spot and droplet center is defined as the optimization metric. The unknown parameters are identified by solving an optimization problem where the target function $\mathcal{J}_{ID}$ is the mean quadratic deviation of the laser-droplet distance between measured and simulated experiments, under the constraints of the system dynamics:

$$\mathcal{J}_{ID}(\mathbf{p}) = \sum_{i=0}^{M} \|d(t_{i,m}) - \hat{d}_i\|^2 \qquad (32)$$

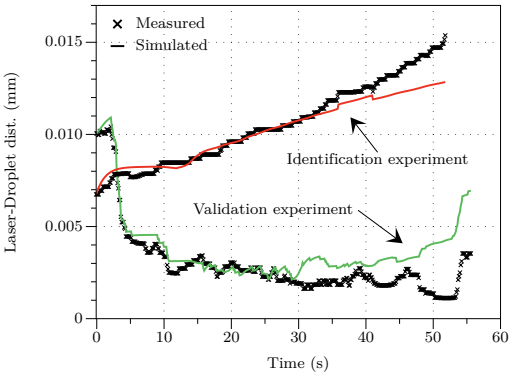


Fig. 9. Comparison of laser-droplet distance evolution for identification and validation experiments, measured and simulated data

Once identified, the model has been used in the second set of experiments for forecasting the droplet motion resulting from a given laser spot trajectory (open loop).

The results of the identification and validation steps are reported in Fig. 9. It may be observed as the laser-droplet distance predicted by the model tracks with good approximation (less than the typical droplet diameter) the actual evolution of the system, at least in the first 30 s. After that time, all the experiments (both in identification and validation conditions) show a diverging drift between measured and simulated values. This is probably related to the delayed effect of convective motion of the fluid body, which has not been included in the model. Nevertheless, with the intention of developing a closed loop, model-predictive control of the laser-induced droplet motion this model inadequacy is of limited impact, since in a receding horizon approach the time horizon of 30 seconds is more than enough for implementing a closed loop control.

V. Conclusions and Future Works

This work describes the conceptual framework and the analytical/numerical approach for developing a motion-planning and control system that can be used for driving a droplet in a 2-D liquid field (e.g. a shallow Petri dish), being the motion driver the Marangoni effect.

The approach consists in a two-steps solution of the problem of finding the optimal time-sequence of laser positions that can move the droplet from point A to point B, complying with the dynamics of the droplet dynamics and of the robotic system moving the laser beam, and subjected to constraints on the initial and final condition of the system. The first step is the calculation of a guess solution by application of the *Model Predictive Control* approach. The second step exploits this guess solution for finding a better one by applying the *Optimal Control* theory.

This approach is applied to two different examples, where the droplet is supposed to be moved in straight line from A to B, where in the first case the droplet velocity in B is unconstrained, while in the second case is constrained to zero. The simulation results show that a rather precise and stable motion control is possible and feasible.

The next step—which is actually an on-going activity—is to use the solution provided by this framework as a control scheme for the robot shown in Fig. 4 thus completing the last dashed box in Fig. 1.

Acknowledgment

M. Hanczyc is financially supported by the European Commission FP7 Future and Emerging Technologies Proactive (EVOBLISS 611640).

References

[1] A. I. Fedosov, "Thermocapillary Motion," *Zhurnal Fizicheskoi Khimii*, vol. 30, no. 2, pp. 366–373, 1956. [Online]. Available: http://arxiv.org/abs/1303.0243

[2] L. H. Dill and R. Balasubramaniam, "Unsteady thermocapillary migration of bubbles," NASA, NASA Technical Report 19890012683, Jan 1988. [Online]. Available: http://ntrs.nasa.gov/search.jsp?R=19890012683

[3] N. Rashidnia and R. Balasubramaniam, "Thermocapillary migration of liquid droplets in a temperature gradient in a density matched system," *Experiments in Fluids*, vol. 11, no. 2–3, pp. 167–174, 1991.

[4] E. Yakhshi-tafti, H. J. Cho, and R. Kumar, "Droplet actuation on a liquid layer due to thermocapillary motion: Shape effect," *Applied Physics Letters*, vol. 96, no. 264101, 2010.

[5] A. Davanlou, R. Shabani, H. J. Cho, and R. Kumar, "Is thermocapillary enough for droplet actuation?" in *17th International Conference on Miniaturized*, no. October. Freiburg, Germany: Systems for Chemistry and Life Sciences, October 2013, pp. 59–61.

[6] A. S. Basu and Y. B. Gianchandani, "Virtual microfluidic traps, filters, channels and pumps using Marangoni flows," *Journal of Micromechanics and Microengineering*, vol. 115031, 2008.

[7] S. Rybalko, N. Magome, and K. Yoshikawa, "Forward and backward laser-guided motion of an oil droplet," *Physical Review E*, vol. 70, no. 046301, pp. 1–4, October 2004.

[8] S.-y. Park and P.-y. Chiou, "Light-Driven Droplet Manipulation Technologies for Lab-on-a-Chip Applications," *Advances in OptoElectronics*, vol. 2011, no. 909174, pp. 1–12, 2011.

[9] D. E. Lucchetta, F. Simoni, L. Nucara, R. Castagna, D. E. Lucchetta, F. Simoni, and L. Nucara, "Controlled-motion of floating macro-objects induced by light Controlled-motion of floating macro-objects induced by light," *AIP Advances*, vol. 077147, no. 2015, 2016. [Online]. Available: http://dx.doi.org/10.1063/1.4927419

[10] M. L. Cordero, D. R. Burnham, C. N. Baroud, and D. McGloin, "Thermocapillary manipulation of droplets using holographic beam shaping: Microfluidic pin ball," *Appl. Phys. Lett.*, vol. 93, no. 034107, 2008.

[11] M. R. de Saint Vincent, R. Wunenburger, and J.-P. Delville, "Laser switching and sorting for high speed digital microfluidics," *Appl. Phys. Lett.*, vol. 92, no. 154105, 2008.

[12] M. M. Hanczyc, "Droplets: Unconventional protocell model with life-like dynamics and room to grow," *Life*, vol. 4, no. 4, pp. 1038–1049, 2014.

[13] M. M. Hanczyc, T. Toyota, T. Ikegami, N. Packard, and T. Sugawara, "Fatty acid chemistry at the oilwater interface: self-propelled oil droplets," *J. Am. Chem. Soc.*, vol. 129, no. 30, pp. 9386–9391, 2007.

[14] J. Čejková, M. Novák, F. Štěpánek, and M. M. Hanczyc, "Dynamics of chemotactic droplets in salt concentration gradients," *Langmuir*, vol. 30, no. 40, pp. 11 937–11 944, 2014.

[15] M. M. Hanczyc, J. M. Parrilla, A. Nicholson, K. Yanev, and K. Stoy, "Creating and maintaining chemical artificial life by robotic symbiosis," *Artif Life*, vol. 21, no. 1, pp. 47–54, 2015.

[16] J. M. P. Gutierrez, T. Hinkley, J. W. Taylor, K. Yanev, and L. Cronin, "Evolution of oil droplets in a chemorobotic platform," *Nature Communications*, vol. 5, no. 5571, 2014.

[17] EVOBLISS 611640 European Commission FP7 Project. [Online]. Available: https://blogit.itu.dk/evoblissproject/

[18] S. L. Anna, "Droplets and bubbles in microfluidic devices," *Annual Review of Fluid Mechanics*, vol. 48, pp. 285–309, 2016.

[19] C. Kunstmann-Olsen, M. Hanczyc, J. Hoyland, S. Rasmussen, and H. Rubahn, "Uniform droplet splitting and detection using lab-on-chip flow cytometry on a microfluidic PDMS device," *Sensors and Actuators B: Chemical*, vol. 229, pp. 7–13, 2016.