ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE - DIN

CORSO DI LAUREA IN INGEGNERIA MECCANICA - 5724

TESI DI LAUREA

in

Meccanica delle Macchine M

TIME-OPTIMAL TRAJECTORY PLANNING FOR ANTI-SLOSHING ROBOTIC MANIPULATION OF MULTIPLE LIQUID-FILLED CONTAINERS

CANDIDATO:
Andrea Ferrari

RELATORE:
Prof. Ing. Marco Carricato

CORRELATORE:
Ph.D. Roberto Di Leva

Anno accademico 2023/2024

Sessione V

A mia nonna Silvana,
so quanto ti sarebbe piaciuto studiare, e spero,
con questo percorso in cui mi hai accompagnato passo dopo passo,
di averti fatto vivere anche solo per una piccolissima parte
quel tuo sogno che è sempre rimasto nel cassetto.

Sommario

Questa tesi esamina tecniche di ottimizzazione vincolata finalizzate a ottenere una manipolazione tempo-ottimale di contenitori riempiti di liquido. In particolare, l'obiettivo è sviluppare algoritmi per la pianificazione di traiettorie tempo-ottimali per più contenitori cilindrici trasportati da un robot industriale. Le traiettorie considerate sono 4-dimensionali, comprendenti 3 dimensioni di traslazione combinate con una rotazione unidimensionale attorno alla direzione verticale.
Quando un contenitore riempito di liquido subisce un moto, il liquido al suo interno esibisce un movimento, un fenomeno noto come *sloshing*. Gli algoritmi proposti si concentrano sul controllo degli effetti dello sloshing durante il movimento del contenitore.
Vengono presentati due approcci di ottimizzazione: nel primo, l'obiettivo è determinare la legge di moto ottimale su un percorso predefinito che il vassoio che ospita i contenitori deve seguire; nel secondo, solo alcuni punti di passaggio, sia con una posizione assegnata sia con una posizione variabile all'interno di un volume assegnato, sono vincolati, mentre i segmenti di percorso rimanenti e la legge di moto sono trattati come variabili nel processo di ottimizzazione. I risultati vengono poi confrontati in simulazione. La procedura più efficace per condurre il processo di ottimizzazione viene identificata, tenendo conto anche della riduzione del tempo di calcolo. Infine, viene realizzata e discussa una campagna di validazione sperimentale per valutare e confrontare l'efficacia pratica di queste tecniche.

Abstract

This thesis investigates constrained-optimization techniques aimed at achieving time-optimal handling of containers filled with liquid. In particular, the objective is to develop algorithms for planning time-optimal trajectories for multiple cylindrical containers carried by an industrial robot. The considered trajectories are 4-dimensional, comprising 3-dimensional translations combined with a 1-dimensional rotation about the vertical direction.
When a container filled with liquid is moved, the liquid inside exhibits motion, a phenomenon known as *sloshing*. The proposed algorithms focus on controlling the effects of sloshing during the container movement.
Two families of optimization approaches are presented: in the first, the objective is to determine the optimal motion law on a predefined path that the tray hosting the multiple containers must follow; in the second, only certain waypoints, either with an assigned position or with a position variable within an assigned volume, are constrained, while the remaining path segments and the motion law are treated as variables in the optimization process. The results are then compared in simulation. The most effective procedure for conducting the optimization process is identified, including considerations for minimizing computational time. Finally, an experimental validation campaign is carried out and discussed to assess and compare the practical effectiveness of these techniques.

Contents

Chapter 1

Introduction

1.1 The sloshing phenomenon

Sloshing refers to the oscillatory motion of a liquid within a partially filled container. This phenomenon plays a crucial role in various fields. In aerospace engineering, the movement of propellant inside fuel tanks can lead to structural and flight-stability issues, requiring advanced control techniques and the design of specific containment baffles [1, 2]. Similarly, in automotive and offshore industries, especially in vehicles used for liquid transport, such as tank trucks and cargo ships, sloshing can generate significant forces due to the liquid oscillatory movement, which impacts vehicle dynamics and must be effectively controlled [3, 4]. In civil engineering, the effect of sloshing inside storage tanks is studied in order to enhance the seismic response of structures, hence using the liquid movement as an energy absorber [5, 6].
This thesis investigates techniques for controlling sloshing in industrial packaging applications. In many processes, it is common to transport bottles, vials, and other containers that are partially filled with liquid. The oscillatory movement of the liquid inside these containers can cause problems, including spillage, inconsistent fill levels, and even structural damage during transit. In the pharmaceutical industry, this phenomenon is particularly critical, as even slight variations in fill levels can affect the dosage and, consequently, the effectiveness of the medication. Moreover, sloshing poses an additional concern in the transport of freeze-dried containers, as liquid residues on the container walls, caused by sloshing, can compromise the freeze-drying process, making it imperative to minimize such disturbances. Similarly, in the food and beverage industry, maintaining product integrity during transport is essential to ensure safety and quality standards. The containers are typically transported in groups by means of an industrial robot, which executes pick-and-place operations from a station to another.

1.2 Targets

The target of this thesis regards the development of techniques aimed at planning pick-and-place trajectories executed by an industrial serial robot, for the simultaneous transport of multiple cylindrical containers. The trajectories must be performed in minimal time, while ensuring that the peak (hereafter referred as *maximum sloshing height*) reached by the liquid inside each container remains under a specified threshold. The considered trajectories are 4-dimensional, with the 3-dimensional translation

combined with a 1-dimensional rotation about a vertical axis. Being mounted on the same rigid tray attached to the robot end-effector, each container is subjected to a different motion depending on the distance from the instantaneous rotation axis. A key constraint in this study is the limitation of rotation only around the vertical axis, since this is a typical constraint in many pick-and-place operations occurring in automated lines; accordingly, the system cannot rely on tilting motions to stabilize the liquid inside the containers.

1.3 State of the art

Before analyzing the sloshing-control techniques that can be retrieved from the literature, it is fundamental to provide an overview of how the sloshing phenomenon can be effectively modeled for these applications. *Computational Fluid Dynamics* (CFD) analyses can be used for this purpose, as detailed in [7] and [8]. These techniques, which solve the Navier–Stokes equations under prescribed boundary conditions, enable the study of complex phenomena such as turbulence, wave breaking, and nonlinear interactions. However, high-fidelity CFD simulations are computationally demanding, often requiring hours or days to complete, especially as domain complexity and resolution increase. Alternatively, methods like *Smoothed Particle Hydrodynamics* (SPH) [9, 10] discretize the fluid into particles to reduce computation time. However, this approach still remains computationally intensive to support real-time applications. Consequently, simplified models have been developed to reproduce sloshing dynamics with considerably lower computational effort. A robust and efficient strategy involves the formulation of equivalent discrete mechanical models, in which the computation of the sloshing height runs in fractions of a second. In [11], two of these discrete mechanical models for describing sloshing dynamics are presented, and they represent the two most widely used models in the literature. The first consists of a spherical pendulum, where angular variables indicating the position of the pendulum mass are used as generalized coordinates to describe the sloshing behavior. This approach is highly effective from an intuitive standpoint (see [12,13]), as the inclination of the free surface of the liquid during motion is represented by the inclination of the pendulum. However, when precise knowledge of the liquid peak is required, this model has limitations. Indeed, in [14], [15], and [16], the sloshing height is estimated by applying a tangent function to the generalized coordinates, and when the dynamic conditions become severe, meaning the container is subjected to high accelerations, the angle may approach 90°, resulting in non-physical outputs from the tangent function. The second model presented in [11], although less intuitive, maintains physical consistency even under high dynamic conditions. For this reason, it is the model adopted in this thesis. This model represents sloshing dynamics through some masses, each representing the modal mass of a sloshing mode, whose motion is governed by a spring-damper system [17]. Using this model, denoted as mass-spring-damper model, the generalized coordinates are represented by the position of the mass with respect to its initial configuration. Based on this model, a novel method for estimating sloshing height was proposed in [18] and validated for 1-dimensional motions. An extension to planar motions is presented in [19], and the experimental verification is provided in [20], where a cylindrical container and accelerations up to $9.5\,\mathrm{m/s^2}$ are considered. An extension to the 3-dimensional case is proposed in [21], which considers an additional vertical acceleration up to $5\,\mathrm{m/s^2}$. The extension of the aforementioned technique to the case

of 4-dimensional motions considering the additional rotation about the vertical axis is object of a concurrent study, whose content is in preparation and will be employed in this thesis, too [22]. Other applications of the mass-spring-damper model can be found in [23] for designing anti-sloshing trajectories and in [24] for a software application that simulates sloshing in both cylindrical and rectangular containers. Regarding sloshing-reduction methods, the literature offers several approaches. Trajectory-planning filters represent a widely used technique to eliminate specific frequency components and reduce vibrations. Several filters can be applied for this purpose, including finite impulse response (FIR) and exponential filters, which are employed in [25] and [26], respectively, and have proven their efficacy in reducing sloshing as well. In [19], a comparison among FIR filters, infinite impulse response (IIR) filters, and dynamic model inversion is provided. A specific class of FIR filters, known as *input shaping* and introduced in [27], is the most recommended one for sloshing-reduction purposes. The most commonly used input shapers are the zero vibration (ZV) shaper, and the zero vibration and zero derivative (ZVD) one: while the former one grants the elimination of residual vibrations as long as the system parameters are precisely known, the latter one offers greater robustness to model uncertainties. An application of this technique is presented in [28], where input shaping is utilized to reduce sloshing in a cylindrical container manipulated by a robotic system along 1-dimensional rectilinear paths, and further applied to 3-dimentional motions in [12]. In the latter study, a shaper called ZV2lin, developed in [29] for tower crane applications, is used to filter rotational motion around the vertical axis. Although shaping techniques are widely used to reduce sloshing, this thesis does not employ them. This decision relied on the fact that, when moving multiple containers simultaneously, each with distinct dynamics, determining the optimal motion profiles on which to apply the filtering becomes both complicated and time-consuming. Furthermore, while these filters can reduce oscillations during the rest phase (i.e., when the container motion is over), they do not constrain the maximum allowable sloshing height during container motion (an aspect which, as discussed in section 1.1, is fundamental to our applications). Additionally, [16] proposes a combination of a *damped harmonic smoother* filter (see [30]) and *tilting*. The latter is another commonly adopted technique to compensate sloshing, as highlighted in [15]; however, as stated in section 1.2, tilting is not allowed in this work. An alternative approach is represented by constrained optimization. In [13], this technique is applied to a single container in a 3-dimensional motion scenario; however, experimental results are provided only for 1-dimensional motions and sloshing is mitigated by reorienting the robot end-effector. In [31], a constrained optimization is used to determine the coefficients that control the profile of a shaper. However, the situation under consideration is quite different from ours, as it focuses on an overhead crane system and does not address the case of multiple containers. In [32], an optimization is performed to design the duration of the segments of a piecewise motion law, but only 1-dimensional trajectories are considered. [12] performs a constrained optimization for generating a trajectory, but sloshing suppression is handled separately using shaping techniques. In [23] and [33], for trajectories with an assigned path, and in [34], for point-to-point motions, constrained optimizations are applied to trajectory planning problems for container transportation in industrial settings. These optimizations are solved using a multiple shooting method and implemented with CasADi [35], a software framework for nonlinear optimization and optimal control in Matlab. This latter technique is the one adopted in this work, with the aim of extending the optimization

to complex 4-dimensional trajectories and multiple containers, an aspect that, to the best of our knowledge, has not been previously addressed in the literature. Indeed, the three articles cited above have the following limitations: [23] and [33] deals with two-dimensional trajectories and only a single container, [34] does not constrain the maximum sloshing height within the optimizer, instead, it determines, using experimental methods and CFD techniques (both of which are time-consuming), the maximum liquid volume that can be safely contained without spillage during the trajectory.

1.4 Thesis structure

In chapter 2, equivalent mechanical models for replicating sloshing dynamics are presented. After highlighting the characteristics of each model and providing their equations of motion, the most suitable models for further analysis are identified. Subsequently, the multi-container configuration is described and the key equations linking sloshing dynamics to each container motion are provided.
Chapter 3 outlines the optimization algorithms employed for time-optimal trajectory planning along predefined paths. Various scenarios are analyzed, and based on their results, the most effective optimization strategies are identified.
Based on these findings, chapter 4 introduces optimization algorithms for point-to-point motion, where the paths are not defined *a priori* but are determined as part of the optimization process, along with the motion law.
Finally, the experimental campaign conducted to validate the developed algorithms is presented in chapter 5. The experimental setup is first described, followed by a detailed analysis of the experimental results.

For the writing of this thesis work, the generative artificial intelligence software ChatGPT-4 was used to support text formulation and ensure grammatical accuracy. Details on its use are provided in appendix D.

Chapter 2

Equivalent mechanical model

In this chapter, several equivalent mechanical models of sloshing dynamics are presented. As discussed in section 1.3, a mass-spring-damper model is adopted; hence, various variants of this model are examined to determine the most appropriate formulation for further study. Each variant considers only the fundamental oscillation mode, since, as demonstrated in [11], higher-order modes have a negligible impact on sloshing behavior. After introducing the models, the ones selected for subsequent applications are identified, the multi-container configuration is presented and the main geometric and kinematic relationships linking the sloshing dynamics of each container to its imposed motion are described.

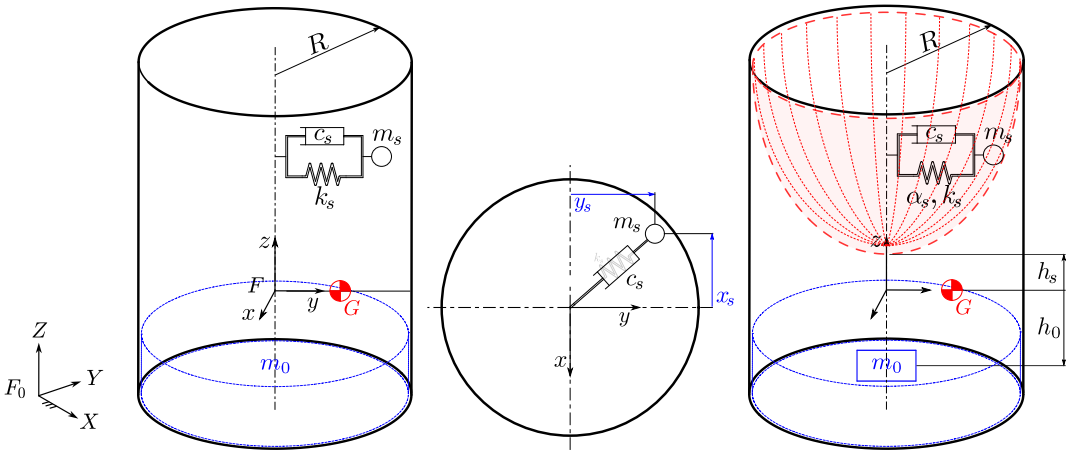
2.1 Model parameters

A cylindrical container with radius R is considered, filled with a liquid of mass m_F. Under static conditions, the liquid height inside the container is h. A fixed coordinate system $F_0 \equiv XYZ$ is defined, along with a moving coordinate system $F \equiv xyz$ attached to the container. The z-axis of F coincides with the container axis and points upward, while the x and y axes pass through the static position of the liquid center of gravity G (see figure 2.1a). The mass-spring-damper model considers a mass m_0 that moves rigidly with the container and a series of masses m_s (each one of them representing the modal mass of a sloshing mode) that move relative to the container (figure 2.1a). Each mass is connected to a spring-damper system that governs its motion. The spring stiffness of the m_s mass is denoted by k_s, while the damping coefficient of the damper connected to the same mass is denoted by c_s. Finally, h_0 and h_s represent the signed vertical distances of m_0 and m_s from G.
To determine the model parameters, equivalences with the continuous model must be imposed, for instance, the total sum of the sloshing masses must be equal to the liquid mass m_F, namely:

$$m_F = m_0 + \sum_{s=1}^{\infty} m_s. \tag{2.1}$$

More details can be found in chapter 5 of [11]. Below, the natural frequency and modal mass of the s-th mode are given:

$$\omega_s = \sqrt{\frac{k_s}{m_s}} = \sqrt{g\frac{\xi_{1s}}{R}\tanh\left(\xi_{1s}\frac{h}{R}\right)}, \tag{2.2}$$



(a) Model Parameters. (b) Top view showing the s-th generalized coordinates. (c) Sloshing mass sliding on a parabolic surface.

Figure 2.1: Mass-spring-damper model.

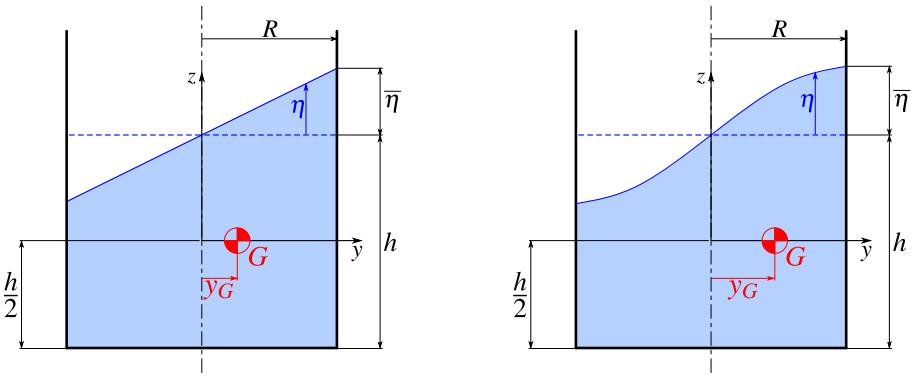
$$m_s = m_F \frac{2R}{\xi_{1s} h(\xi_{1s}^2 - 1)} \tanh\left(\xi_{1s} \frac{h}{R}\right). \tag{2.3}$$

In (2.3) the parameter ξ_{1s} represents the root of the derivative of the Bessel function of the first kind [36], and g is the gravity acceleration. The damping ratio $\zeta_s = \dfrac{c_s}{2\sqrt{k_s m_s}}$ can be determined by using the following experimental formula (see [11]):

$$\zeta_s = 0.92 \sqrt{\frac{\upsilon/\rho}{\sqrt{gR^3}}} \left[1 + \frac{0.318}{\sinh(\xi_{1s} h/R)} \left(1 + \frac{1 - h/R}{\cosh(\xi_{1s} h/R)}\right)\right], \tag{2.4}$$

where υ and ρ are the dynamic viscosity and density of the liquid, respectively.

From [11], for a liquid inside a cylindrical container, three dynamic regimes can occur depending on the excitations applied to the container. The first regime, depicted in figure 2.2a, is characterized by a planar liquid surface, which occurs under small oscillations. When the liquid oscillations increase moderately, the system enters the second regime, in which the surface no longer maintains a planar shape, this scenario is illustrated in figure 2.2b. Finally, when the motion exhibits strong nonlinearities,



(a) Planar free surface. (b) Nonplanar free surface.

Figure 2.2: Liquid free-surface shapes.

the liquid surface develops instantaneous peaks. This third scenario is not addressed in this thesis. For the first two cases, two models will be developed in the following sections to represent the liquid dynamics: one linear and one nonlinear. The linear model represents the condition of a planar liquid surface, considers the spring stiffness as a constant, and neglects the vertical motion of G. As a result, the mass m_s does not change its vertical component during motion, which remains constant at $z_s = h_s$, where h_s is given by [11]:

$$h_s = \frac{1}{2}h\Big[1 - \frac{4R\tanh(\xi_{1s}\frac{h}{2R})}{\xi_{1s}h}\Big]. \tag{2.5}$$

The non-linear model, on the other hand, considers the case of a non-planar liquid surface, which is described by a first-kind Bessel function [21]. In this case, the sloshing mass m_s slides on a parabolic surface and is restrained by a nonlinear spring of order w (figure 2.1c) [17]. Therefore, differently from the linear case, there is motion along z, which can be expressed as a function of x_s and y_s, namely:

$$z_s = h_s + \frac{C_s}{2R}(x_s^2 + y_s^2), \tag{2.6}$$

where

$$C_s = \xi_{1s}\tanh\Big(\xi_{1s}\frac{h}{R}\Big) = \omega_s^2\frac{R}{g}. \tag{2.7}$$

The velocity of m_s along z is obtained by differentiating x.r.t. time the equation (2.6), resulting in:

$$\dot{z}_s = \frac{C_s}{R}(\dot{x}_s x_s + \dot{y}_s y_s). \tag{2.8}$$

The nonlinear spring exerts a force given by $\alpha_s k_s r_s^{2w-1}$, where r_s is given by: $r_s = \sqrt{x_s^2 + y_s^2}$; and the parameters w and α_s, as suggested in [17], are set as follows: $w = 2$, $\alpha_s = 0.58$.
In conclusion, both in the case of the linear model and in the case of the nonlinear model, the motion of the mass m_s is fully described by the parameters x_s and y_s (figure 2.1b), which will therefore be used as generalized coordinates in the Lagrangian system of the next section.

2.2 Equations of motion

In this thesis, a 4-dimensional motion is considered, comprising translation along the three directions and rotation around a vertical axis. The excitations to which the container, and therefore the liquid, is subjected can be defined as follows: $\ddot{\mathbf{S}}_0 = [\ddot{x}_0 \;\; \ddot{y}_0 \;\; \ddot{z}_0]^T$ for linear motions, and $\dot{\mathbf{\Omega}}_0 = [0 \;\; 0 \;\; \ddot{\theta}_0]^T$ for rotational motions. The subscript 0 indicates that the quantities are expressed w.r.t. the fixed reference frame. The vector $\mathbf{S}_0$ is the position of the centerpoint O of the container bottom circle, while θ_0 is the rotation of the container around the Z axis.
To obtain the equations of motion (EOMs), the Lagrange equations are used. As mentioned at the end of section 2.1, the used generalized coordinates, (q_1, q_2), are (x_s, y_s). The Lagrange equations are:

$$\frac{d}{dt}\Big(\frac{\partial T}{\partial \dot{q}_i}\Big) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} = 0, \quad \text{for } i = 1,2 \tag{2.9}$$

where T is the kineic energy, V is the potential energy and D is the Rayleigh function. The position vector $\mathbf{r}_s$ of the s-th sloshing mass can be written as:

$$ {}^0\mathbf{r}_s = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \mathbf{R}_z(\theta_l) \begin{bmatrix} x_s \\ y_s \\ \frac{h}{2} + z_s \end{bmatrix} = \begin{bmatrix} x_0 + x_s\cos\theta_l - y_s\sin\theta_l \\ y_0 + x_s\sin\theta_l + y_s\cos\theta_l \\ z_0 + \frac{h}{2} + h_s + \dfrac{C_s}{2R}(x_s^2 + y_s^2) \end{bmatrix}, \tag{2.10} $$

where $\mathbf{R}_z(\theta_l)$ is the rotation matrix expressing the rotation of angle θ_l about the vertical z-axis, namely:

$$ \mathbf{R}_z(\theta_l) = \begin{bmatrix} \cos\theta_l & -\sin\theta_l & 0 \\ \sin\theta_l & \cos\theta_l & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.11} $$

By differentiating equation (2.10) w.r.t. time, we obtain:

$$ {}^0\dot{\mathbf{r}}_s = \begin{bmatrix} \dot{x}_0 + \dot{x}_s\cos\theta_l - x_s\dot{\theta}_l\sin\theta_l - \dot{y}_s\sin\theta_l - y_s\dot{\theta}_l\cos\theta_l \\ \dot{y}_0 + \dot{x}_s\sin\theta_l + x_s\dot{\theta}_l\cos\theta_l + \dot{y}_s\cos\theta_l - y_s\dot{\theta}_l\sin\theta_l \\ \dot{z}_0 + \dfrac{C_s}{R}(\dot{x}_s x_s + \dot{y}_s y_s) \end{bmatrix}. \tag{2.12} $$

The total kinetic energy can be expressed as the sum of the kinetic energies of the individual sloshing masses. This takes into account the velocities of the sloshing masses relative to the container, $\dot{\mathbf{s}}_s = [\dot{x}_s \;\; \dot{y}_s \;\; \dot{z}_s]^T$, and the velocities of the container, $\dot{\mathbf{S}}_0$ and $\dot{\mathbf{\Omega}}_0$:

$$ \begin{aligned} T &= T_0 + \sum_s T_s = \\ &= T_0 + \sum_s \frac{1}{2} m_s \Big[\dot{x}_0^2 + \dot{y}_0^2 + \dot{z}_0^2 + \dot{x}_s^2 + \dot{y}_s^2 + \frac{C_s^2}{R^2}\left(x_s\dot{x}_s + y_s\dot{y}_s\right)^2 + \dot{\theta}_l^2\left(x_s^2 + y_s^2\right) + \\ &+ 2\dot{z}_0\frac{C_s}{R}(x_s\dot{x}_s + y_s\dot{y}_s) - 2\dot{x}_s y_s\dot{\theta}_l + 2\dot{y}_s x_s\dot{\theta}_l + \\ &+ 2(\dot{x}_0\cos\theta_l + \dot{y}_0\sin\theta_l)\dot{x}_s + 2(-\dot{x}_0\dot{\theta}_l\sin\theta_l + \dot{y}_0\dot{\theta}_l\cos\theta_l)x_s + \\ &+ 2(-\dot{x}_0\sin\theta_l + \dot{y}_0\cos\theta_l)\dot{y}_s + 2(-\dot{x}_0\dot{\theta}_l\cos\theta_l - \dot{y}_0\dot{\theta}_l\sin\theta_l)y_s \Big], \end{aligned} \tag{2.13} $$

where T_0 is the kinetic energy of the mass m_0 (which does not depend on x_s and y_s) and T_s is the kinetic energy of the s-th sloshing mass.
The potential energy V can be written in the same way; it considers the contribution of gravity and non-linear spring forces:

$$ \begin{aligned} V &= V_0 + \sum_s V_s = V_0 + \sum_s \Big(m_s g\big(z_0 + \frac{h}{2} + z_s\big) + \int_0^{r_s} \alpha_s k_s r_s^{2w-1} dr_s \Big) = \\ &= V_0 + \sum_s \Big\{ m_s g\Big[z_0 + \frac{h}{2} + h_s + \frac{C_s}{2R}(x_s^2 + y_s^2) \Big] + \frac{\alpha_s k_s}{2w}(x_s^2 + y_s^2)^w \Big\}, \end{aligned} \tag{2.14} $$

where V_0 is the potential energy of m_0 (which does not depend on x_s and y_s) and V_s is the potential energy of the s-th sloshing mass.
The Rayleigh function D considers energy dissipation and it only depends on the sloshing-mass velocity $\dot{\mathbf{s}}_s = [\dot{x}_s \;\; \dot{y}_s \;\; \dot{z}_s]^T$ with respect to the container (the dissipation associated with the viscous-dragging phenomenon discussed in section 2.3.2 is neglected):

$$ D = \sum_s \frac{1}{2} c_s(\dot{x}_s^2 + \dot{y}_s^2 + \dot{z}_s^2) = \sum_s m_s\zeta_s\omega_s \Big[\dot{x}_s^2 + \dot{y}_s^2 + \frac{C_s^2}{R^2}(\dot{x}_s x_s + \dot{y}_s y_s)^2 \Big]. \tag{2.15} $$

Substituting equations (2.13), (2.14) and (2.15) in the system (2.9) leads to two coupled EOMs for the non-linear model (described in section 2.1):

$$\begin{cases} \left(1+\frac{C_s^2}{R^2}x_s^2\right)\ddot{x}_s+\frac{C_s^2}{R^2}x_s y_s \ddot{y}_s = -\frac{C_s^2}{R^2}(\dot{x}_s^2+\dot{y}_s^2)x_s+(2\dot{\theta}_l\dot{y}_s+\dot{\theta}_l^2 x_s+\ddot{\theta}_l y_s) \\ \qquad -\omega_s^2 x_s\left[1+\alpha_s(x_s^2+y_s^2)^{w-1}\right]-2\omega_s\zeta_s[\dot{x}_s+\frac{C_s^2}{R^2}(x_s\dot{x}_s+y_s\dot{y}_s)x_s] \\ \qquad -\ddot{x}_0\cos\theta_l-\ddot{y}_0\sin\theta_l-\ddot{z}_0\frac{C_s}{R}x_s \\ \\ \left(1+\frac{C_s^2}{R^2}y_s^2\right)\ddot{y}_s+\frac{C_s^2}{R^2}x_s y_s \ddot{x}_s = -\frac{C_s^2}{R^2}(\dot{x}_s^2+\dot{y}_s^2)y_s+(-2\dot{\theta}_l\dot{x}_s+\dot{\theta}_l^2 y_s-\ddot{\theta}_l x_s) \\ \qquad -\omega_s^2 y_s\left[1+\alpha_s(x_s^2+y_s^2)^{w-1}\right]-2\omega_s\zeta_s[\dot{y}_s+\frac{C_s^2}{R^2}(x_s\dot{x}_s+y_s\dot{y}_s)y_s] \\ \qquad +\ddot{x}_0\sin\theta_l-\ddot{y}_0\cos\theta_l-\ddot{z}_0\frac{C_s}{R}y_s \end{cases} \tag{2.16}$$

Eliminating the nonlinear terms w.r.t. the generalized coordinates and their derivatives $[x_s\ y_s\ \dot{x}_s\ \dot{y}_s\ \ddot{x}_s\ \ddot{y}_s]$, the system of linearized EOMs is obtained[1]:

$$\begin{cases} \ddot{x}_s = (2\dot{\theta}_l\dot{y}_s+\dot{\theta}_l^2 x_s+\ddot{\theta}_l y_s)-\omega_s^2 x_s-2\omega_s\zeta_s\dot{x}_s-\ddot{x}_0\cos\theta_l-\ddot{y}_0\sin\theta_l-\ddot{z}_0\frac{C_s}{R}x_s \\ \\ \ddot{y}_s = (-2\dot{\theta}_l\dot{x}_s+\dot{\theta}_l^2 y_s-\ddot{\theta}_l x_s)-\omega_s^2 y_s-2\omega_s\zeta_s\dot{y}_s+\ddot{x}_0\sin\theta_l-\ddot{y}_0\cos\theta_l-\ddot{z}_0\frac{C_s}{R}y_s \end{cases} \tag{2.17}$$

The EOMs in (2.17) are linear w.r.t. the state variables, and are thus easier and quicker to solve.

2.3 Sloshing-height estimation

In the previous section, the equations to determine, starting from the imposed accelerations on the container $\ddot{\mathbf{S}}_0$ and $\dot{\mathbf{\Omega}}_0$, the generalized coordinates x_s and y_s were derived. In particular, the equations (2.16) represent the nonlinear model, while the equations (2.17) represent the linearized model. From the knowledge of the generalized coordinates, for both the linearized and non-linear cases, it is possible to derive the maximum sloshing height (referred to, from now on, as $\overline{\eta}_s$) of the s-th mode, based on the results presented in [21], as shown below:

- using the generalized coordinates obtained from the non-linear EOMs (2.16):

$$\overline{\eta}_s = \frac{\xi_{1s}^2 h m_s}{m_F R}\sqrt{x_s^2+y_s^2}, \tag{2.18}$$

- using the generalized coordinates obtained from the linearized EOMs (2.17):

$$\overline{\eta}_s = \frac{4 h m_s}{m_F R}\sqrt{x_s^2+y_s^2}. \tag{2.19}$$

The total sloshing height can be then formulated as the combination of all the s modes considered. However, it can be shown [18] that the contributions of modes greater

[1] It should be noted that the system of equations does not represent the previously described linear model, as that model neglects the motion of G along the z-axis. Therefore, this model should be understood as a linearized version of the non-linear one, making it easier to solve.

than the first one are often negligible, hence justifying the only employment of the first sloshing mode. In [21], the rotational motion was not considered, and therefore, the two EOMs, both for the linearized and nonlinear cases, were functions of only the two unknowns x_s and y_s, and could therefore be solved directly. In our case, the equations have an additional unknown, which is the rotation of the liquid around the vertical axis, θ_l. It is therefore necessary to introduce new equations or assumptions in order to solve the two equations of motion with the three unknowns x_s, y_s, and θ_l. Below, three different methods are presented to establish new relationships between the variable θ_l and the motion conditions imposed on the container.

2.3.1 Sloshing model considering $\theta_l = \theta_0$

In this model, referred to as NL if EOMs (2.16) are considered and the condition $\theta_l = \theta_0$ is enforced or L if EOMs (2.17) are considered and the condition $\theta_l = \theta_0$ is imposed, the EOMs are solved by simply assuming that the rotation of the liquid θ_l is equal to the rotation of the container θ_0, the latter being known *a priori* as an input to the system. Thus, the following equations are obtained.

- For the NL model:

$$\begin{cases} \left(1+\frac{C_s^2}{R^2}x_s^2\right)\ddot{x}_s + \frac{C_s^2}{R^2}x_s y_s \ddot{y}_s = -\frac{C_s^2}{R^2}(\dot{x}_s^2+\dot{y}_s^2)x_s + (2\dot{\theta}_0\dot{y}_s + \dot{\theta}_0^2 x_s + \ddot{\theta}_0 y_s) \\ \qquad -\omega_s^2 x_s\left[1+\alpha_s(x_s^2+y_s^2)^{w-1}\right] - 2\omega_s\zeta_s[\dot{x}_s + \frac{C_s^2}{R^2}(x_s\dot{x}_s + y_s\dot{y}_s)x_s] \\ \qquad -\ddot{x}_0\cos\theta_0 - \ddot{y}_0\sin\theta_0 - \ddot{z}_0\frac{C_s}{R}x_s \\ \\ \left(1+\frac{C_s^2}{R^2}y_s^2\right)\ddot{y}_s + \frac{C_s^2}{R^2}x_s y_s \ddot{x}_s = -\frac{C_s^2}{R^2}(\dot{x}_s^2+\dot{y}_s^2)y_s + (-2\dot{\theta}_0\dot{x}_s + \dot{\theta}_0^2 y_s - \ddot{\theta}_0 x_s) \\ \qquad -\omega_s^2 y_s\left[1+\alpha_s(x_s^2+y_s^2)^{w-1}\right] - 2\omega_s\zeta_s[\dot{y}_s + \frac{C_s^2}{R^2}(x_s\dot{x}_s + y_s\dot{y}_s)y_s] \\ \qquad +\ddot{x}_0\sin\theta_0 - \ddot{y}_0\cos\theta_0 - \ddot{z}_0\frac{C_s}{R}y_s \end{cases} \tag{2.20}$$

  The two equations only involve the two unknowns x_s and y_s; therefore, they can be solved. The maximum sloshing height is determined through the equation (2.18), in which only the first sloshing mode is considered:

$$\overline{\eta}_{NL} = \frac{\xi_{11}^2 h m_1}{m_F R}\sqrt{x_1^2+y_1^2}. \tag{2.21}$$

- For the L model:

$$\begin{cases} \ddot{x}_s = (2\dot{\theta}_0\dot{y}_s + \dot{\theta}_0^2 x_s + \ddot{\theta}_0 y_s) - \omega_s^2 x_s - 2\omega_s\zeta_s\dot{x}_s - \ddot{x}_0\cos\theta_0 - \ddot{y}_0\sin\theta_0 - \ddot{z}_0\frac{C_s}{R}x_s \\ \\ \ddot{y}_s = (-2\dot{\theta}_0\dot{x}_s + \dot{\theta}_0^2 y_s - \ddot{\theta}_0 x_s) - \omega_s^2 y_s - 2\omega_s\zeta_s\dot{y}_s + \ddot{x}_0\sin\theta_0 - \ddot{y}_0\cos\theta_0 - \ddot{z}_0\frac{C_s}{R}y_s \end{cases} \tag{2.22}$$

  The two equations only involve the two unknowns x_s and y_s; therefore, they can be solved. The maximum sloshing height is determined through the equation (2.19):

$$\overline{\eta}_L = \frac{4 h m_1}{m_F R}\sqrt{x_1^2+y_1^2}. \tag{2.23}$$

2.3.2 Sloshing model with the viscous-dragging approach

In the viscous-dragging approach, the sloshing masses are assumed to move on a liquid 'disk' spinning with angular velocity $\dot{\theta}_l$, so that each sloshing mass is subjected to a centripetal acceleration proportional to $\dot{\theta}_l^2$. The relation between the angular velocity of the liquid 'disk', $\dot{\theta}_l$, and the angular velocity of the container, $\dot{\theta}_0$, is inferred by considering the former dragged by the latter through a viscous torque proportional to their relative velocity, namely:

$$J\ddot{\theta}_l = k_{vd}\left(\dot{\theta}_0 - \dot{\theta}_l\right), \tag{2.24}$$

where J is the moment of inertia of the liquid w.r.t. the axis of the container (assumed constant), and k_{vd} is the viscous friction parameter, which depends on the properties of the fluid and the container. For determining this coefficient, there is limited literature, with some references regarding the fluid delay time in reaching $95-99\%$ of the container velocity. For example, in [37], an equation is provided to determine this time in the case of a container starting from rest and reaching a certain angular velocity:

$$t_{99\%} = \frac{R^2}{\nu}, \tag{2.25}$$

where ν is the kinematic viscosity of the fluid.
By using equation (2.25), and integrating equation (2.24), it is possible to determine an approximate value of k_{vd}[2].
Based on these considerations, the equations of the models with a viscous-dragging approach are reported below, referred to as: NL_{vd} and L_{vd}.

- The NL_{vd} model is given by the following equations:

$$\begin{cases} \left(1+\frac{C_s^2}{R^2}x_s^2\right)\ddot{x}_s + \frac{C_s^2}{R^2}x_s y_s \ddot{y}_s = -\frac{C_s^2}{R^2}(\dot{x}_s^2+\dot{y}_s^2)x_s + (2\dot{\theta}_l\dot{y}_s + \dot{\theta}_l^2 x_s + \ddot{\theta}_l y_s) \\ \qquad -\omega_s^2 x_s\left[1+\alpha_s(x_s^2+y_s^2)^{w-1}\right] - 2\omega_s\zeta_s[\dot{x}_s + \frac{C_s^2}{R^2}(x_s\dot{x}_s + y_s\dot{y}_s)x_s] \\ \qquad -\ddot{x}_0\cos\theta_l - \ddot{y}_0\sin\theta_l - \ddot{z}_0\frac{C_s}{R}x_s \\ \\ \left(1+\frac{C_s^2}{R^2}y_s^2\right)\ddot{y}_s + \frac{C_s^2}{R^2}x_s y_s \ddot{x}_s = -\frac{C_s^2}{R^2}(\dot{x}_s^2+\dot{y}_s^2)y_s + (-2\dot{\theta}_l\dot{x}_s + \dot{\theta}_l^2 y_s - \ddot{\theta}_l x_s) \\ \qquad -\omega_s^2 y_s\left[1+\alpha_s(x_s^2+y_s^2)^{w-1}\right] - 2\omega_s\zeta_s[\dot{y}_s + \frac{C_s^2}{R^2}(x_s\dot{x}_s + y_s\dot{y}_s)y_s] \\ \qquad +\ddot{x}_0\sin\theta_l - \ddot{y}_0\cos\theta_l - \ddot{z}_0\frac{C_s}{R}y_s \\ \\ J\ddot{\theta}_l = k_{vd}\left(\dot{\theta}_0 - \dot{\theta}_l\right) \end{cases} \tag{2.26}$$

  The system consists of three equations with three unknowns: x_s, y_s, and θ_l, which can, therefore, be solved. The generalized coordinates obtained can then be used to determine the maximum sloshing height using equation (2.18):

$$\overline{\eta}_{NL,vd} = \frac{\xi_{11}^2 h m_1}{m_F R}\sqrt{x_1^2+y_1^2}. \tag{2.27}$$

[2] To avoid redundancy, further analyses regarding the determination of the parameter k_{vd} are not provided, since this model was not used for the subsequent analyses.

- The L_{vd} model is given by the following equations:

$$\begin{cases} \ddot{x}_s = (2\dot{\theta}_l \dot{y}_s + \dot{\theta}_l^2 x_s + \ddot{\theta}_l y_s) - \omega_s^2 x_s - 2\omega_s \zeta_s \dot{x}_s - \ddot{x}_0 \cos\theta_l - \ddot{y}_0 \sin\theta_l - \ddot{z}_0 \frac{C_s}{R} x_s \\ \ddot{y}_s = (-2\dot{\theta}_l \dot{x}_s + \dot{\theta}_l^2 y_s - \ddot{\theta}_l x_s) - \omega_s^2 y_s - 2\omega_s \zeta_s \dot{y}_s + \ddot{x}_0 \sin\theta_l - \ddot{y}_0 \cos\theta_l - \ddot{z}_0 \frac{C_s}{R} y_s \\ J\ddot{\theta}_l = k_{vd}\left(\dot{\theta}_0 - \dot{\theta}_l\right) \end{cases} \tag{2.28}$$

  As with the NL_{vd} model, we have three equations with three unknowns. The generalized coordinates obtained from the system can be used to determine the maximum sloshing height using equation (2.19):

$$\overline{\eta}_{L,vd} = \frac{4hm_1}{m_F R}\sqrt{x_1^2 + y_1^2}. \tag{2.29}$$

2.3.3 Sloshing model with forced-vortex motion approach

Forced-vortex motion refers to the scenario in which a container filled with liquid undergoes pure rotational motion around its vertical axis. Under these motion conditions, the liquid surface assumes the shape of a paraboloid centered on the rotation axis. Therefore, considering a cylindrical container rotating around its axis at a constant angular velocity (i.e., $\ddot{\theta}_0 = 0$ and $\dot{\theta}_0 \neq 0$) and without translation, referring to [38], the following equation for the surface of the paraboloid (which refers to figure 2.3) can be written:

$$z_s(r) = h_c + \frac{\dot{\theta}_0^2 r^2}{2g}, \tag{2.30}$$

where h_c represents the vertical height of the paraboloid vertex, which is the lowest point of the liquid surface. The other term of equation 2.30 represents the contribution to the height, as a function of r, that is added due to the parabolic profile. Looking at figure 2.3, it is clear that the maximum height is assumed by the liquid in contact with the walls, and it is obtained by substituting $r = R$ into (2.30), yielding:

$$z_s(R) = h_c + \frac{\dot{\theta}_0^2 R^2}{2g}. \tag{2.31}$$

Therefore, it is possible to determine the maximum sloshing height of the liquid under this type of motion, that is, the difference between the highest point of the liquid $z_s(R)$ and the static liquid height h, called $\overline{\eta}_p$:

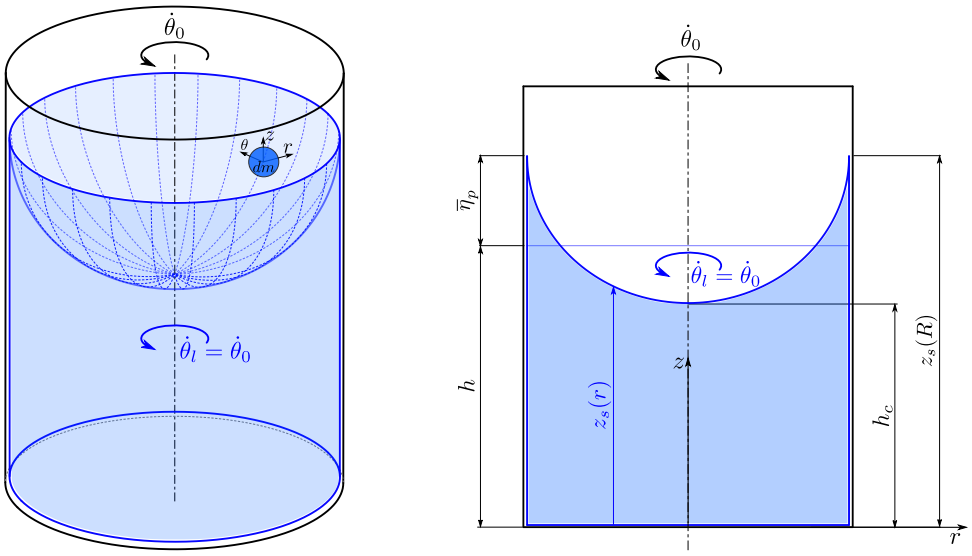
$$\overline{\eta}_p = z_s(R) - h = (h_c - h) + \frac{\dot{\theta}_0^2 R^2}{2g}. \tag{2.32}$$

The volume of the liquid below the paraboloid is:

$$V = \int_0^R 2\pi z_s(r) r\,dr = \int_0^R 2\pi \left(h_c + \frac{\dot{\theta}_0^2 r^2}{2g}\right) r\,dr = \pi R^2 \left(\frac{\dot{\theta}_0^2 R^2}{4g} + h_c\right). \tag{2.33}$$

The liquid volume must be conserved, therefore, by equating (2.33) to the expression for the volume at rest, $V_0 = \pi R^2 h$, the following relationship can be obtained:

$$(h_c - h) = -\frac{\dot{\theta}_0^2 R^2}{4g}. \tag{2.34}$$



(a) 3-dimensional view. (b) Front view.

Figure 2.3: Schematic of a liquid under forced-vortex motion in a cylindrical container.

Substituting equation (2.34) into (2.32), the expression for the contribution of the paraboloid to the sloshing height exclusively as a function of the angular velocity $\dot{\theta}_0$ is obtained:

$$\overline{\eta}_p = \frac{\dot{\theta}_0{}^2 R^2}{4g}. \tag{2.35}$$

Starting from the presented results, the models addressed in this section are referred to as NL_p for the non-linear case and L_p for the linearized case. The underlying idea of these models is to determine the maximum sloshing height using the superposition principle, i.e., by summing the maximum sloshing height obtained considering only translational motion and the maximum sloshing height obtained considering a steady-state rotational motion. Therefore, the EOMs (2.16) (for the non-linear case) and (2.17) (for the linearized case) are solved considering only translational motion (i.e., imposing $\theta_l = \dot{\theta}_l = \ddot{\theta}_l = 0$), thereby determining the sloshing height due only to translation. The sloshing height obtained with the parabolic term from equation (2.35), which only considers rotation, is then added. The models are therefore presented below.

- For the NL_p model, the motion equations are obtained by setting $\theta_l = \dot{\theta}_l = \ddot{\theta}_l = 0$ in equation (2.16):

$$\begin{cases} \left(1 + \frac{C_s^2}{R^2} x_s^2\right) \ddot{x}_s + \frac{C_s^2}{R^2} x_s y_s \ddot{y}_s = -\frac{C_s^2}{R^2} (\dot{x}_s^2 + \dot{y}_s^2) x_s - \omega_s^2 x_s \left[1 + \alpha_s (x_s^2 + y_s^2)^{w-1}\right] \\ \qquad -2\omega_s \zeta_s [\dot{x}_s + \frac{C_s^2}{R^2} (x_s \dot{x}_s + y_s \dot{y}_s) x_s] - \ddot{x}_0 - \ddot{z}_0 \frac{C_s}{R} x_s \\ \left(1 + \frac{C_s^2}{R^2} y_s^2\right) \ddot{y}_s + \frac{C_s^2}{R^2} x_s y_s \ddot{x}_s = -\frac{C_s^2}{R^2} (\dot{x}_s^2 + \dot{y}_s^2) y_s - \omega_s^2 y_s \left[1 + \alpha_s (x_s^2 + y_s^2)^{w-1}\right] \\ \qquad -2\omega_s \zeta_s [\dot{y}_s + \frac{C_s^2}{R^2} (x_s \dot{x}_s + y_s \dot{y}_s) y_s] - \ddot{y}_0 - \ddot{z}_0 \frac{C_s}{R} y_s \end{cases} \tag{2.36}$$

The two equations involve only the two unknowns x_s and y_s, and, therefore, can be solved. From the obtained coordinates x_s and y_s, the maximum sloshing height is determined through equation (2.18), to which, by superposition of

effects, the term $\overline{\eta}_p$ from equation (2.35) is added:

$$\overline{\eta}_{NL,p} = \frac{\xi_{11}^2 h m_1}{m_F R}\sqrt{x_1^2 + y_1^2} + \frac{\dot{\theta}_0^2 R^2}{4g}. \tag{2.37}$$

For the L_p model, the motion equations are obtained by setting $\theta_l = \dot{\theta}_l = \ddot{\theta}_l = 0$ in equation (2.17):

$$\begin{cases} \ddot{x}_s = -\omega_s^2 x_s - 2\omega_s \zeta_s \dot{x}_s - \ddot{x}_0 - \ddot{z}_0 \frac{C_s}{R} x_s \\ \\ \ddot{y}_s = -\omega_s^2 y_s - 2\omega_s \zeta_s \dot{y}_s - \ddot{y}_0 - \ddot{z}_0 \frac{C_s}{R} y_s \end{cases} \tag{2.38}$$

The two equations involve only the two unknowns x_s and y_s, and, therefore, can be solved. From the obtained coordinates x_s and y_s, the maximum sloshing height is determined through equation(2.19), to which, by superposition of effects, the term $\overline{\eta}_p$ from equation (2.35) is added:

$$\overline{\eta}_{L,p} = \frac{4 h m_1}{m_F R}\sqrt{x_1^2 + y_1^2} + \frac{\dot{\theta}_0^2 R^2}{4g}. \tag{2.39}$$

2.4 Equivalent mechanical models used for optimizations

Among all the models discussed, two different models have been used for the subsequent analyses:

- The linearized model with parabolic contribution, denoted as L_p[3], of section 2.3.3, is used in chapters 3 and 4. This model, which presents linearized EOMs (2.38), is simpler and faster for the optimizer to solve compared to the non-linear one (2.16). In the two aforementioned chapters, the objective is to determine the best strategies for performing simulations. For this purpose, numerous scenarios were compared to identify the most effective choices. To report a significant number of results for each scenario, a large number of optimizations were performed. Therefore, to accelerate the process, the use of a simpler model proved to be the best choice. Regarding the addition of the parabolic term in the calculation of the sloshing height, this was included for two main reasons. The first reason is that, in this thesis, we aim to determine the best optimization strategies for 4-dimensional motions, which also consider the effect of the container rotation θ_0 around its vertical axis. However, for the considered motions and the small dimensions of the used containers, the effect of this rotation is barely noticeable using the viscous-dragging approach (section 2.3.2). Therefore, the addition of the parabolic term makes the effect of rotation more significant on the optimization results, allowing us to clearly observe its effect and tailor our strategies to effectively integrate this component. The second reason for adding the parabolic term is that it slightly overestimates the sloshing height. Since the L_p model provides a less accurate estimation compared with the non-linear one, the parabolic term provides a safety margin to the solution, mitigating the deficiencies caused by the approximation of the L_p model and ensuring compliance with the maximum imposed sloshing-height limits;

[3] As discussed, in this model the liquid rotation θ_l is considered equal to zero.

- The non-linear model considering the liquid rotation θ_l equal to the container rotation θ_0, denoted as NL, of section 2.3.1, is used in chapter 5, where the algorithms for the experimental tests are developed. It is adopted because it has proven to be the most accurate in modeling the liquid dynamics for this type of applications, thus making it the most suitable choice for developing trajectories aimed at experimental validation.

An example of the sloshing-height trends predicted by the two models mentioned above, compared with the experimental trend, is shown in figure 2.4. As explained in the previous chapter, it is recalled that the sloshing-height trend related to the L_p model is obtained from equation 2.39, comprising the generalized coordinates derived from the linearized equations in which the liquid rotation is set to zero (equations (2.38)), and the term related to the paraboloid (2.35). On the other hand, the trend of $\overline{\eta}$ for the NL model is obtained from equation (2.21), where x_s and y_s are obtained by solving the general equations (2.16), in which the liquid rotation θ_l is considered equal to that of the container θ_0. It can be observed that the NL model is more accurate, while the L_p model overestimates the values of $\overline{\eta}$, in accordance with the previous discussion. In any case, the NL model still provides an overestimation of the maximum value of $\overline{\eta}$ and this ensures that, even employing the NL model in the optimization, the obtained results will likely remain on the side of safety.

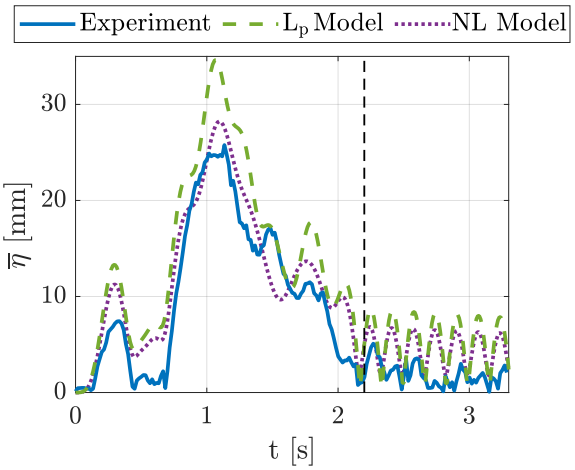


Figure 2.4: Comparison between sloshing models.

Since in both models used, the liquid rotation θ_l is assumed to be equal to the container rotation θ_0, from now on, reference will be made solely to θ as the vertical rotation angle of both the liquid and the container.

2.5 Multi-container configuration

Given a generic number of containers mounted in a row on a rigid support, let $Oxyz$ be a fixed reference frame and $O'x'y'z'$ a reference frame attached to the tray hosting the containers. Recalling that, in our analysis, the containers always maintain their axis oriented vertically, the z' axis is always parallel to the z axis, and θ is defined as the angle between the x' axis and the x axis. Figure 2.5 illustrates this configuration.
Considering the i-th container[4], let O_i (which lies on the x' axis) be the intersection

[4]For clarity, it is important to emphasize that all the containers are identical and filled with the same amount of liquid, consequently, they have the same radius, liquid static height, natural frequency and

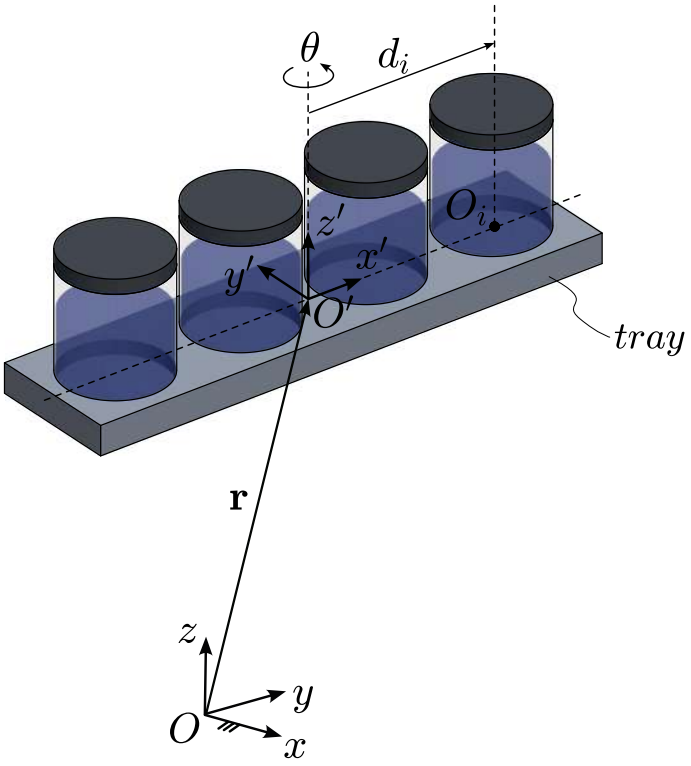


Figure 2.5: Multi-container configuration.

point between the base cirle and its axis. The position vector between O_i and O' can be expressed w.r.t. the moving reference system as:

$$'\mathbf{d}_i = '\left(O_i - O'\right) = \begin{bmatrix} d_i \\ 0 \\ 0 \end{bmatrix}, \tag{2.40}$$

where d_i is the signed distance from the z' axis to the i-th container axis.
Let $\mathbf{r}$ be the position vector of O' w.r.t. the fixed reference system. The position of O_i w.r.t. the fixed system can be written as:

$$\mathbf{r}_i = \mathbf{r} + \mathbf{R}_z(\theta)'\mathbf{d}_i = \mathbf{r} + \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_i \\ 0 \\ 0 \end{bmatrix} = \mathbf{r} + \mathbf{d}_i. \tag{2.41}$$

By differentiating (2.41) w.r.t. time, the velocity and acceleration of the i-th container, according to rigid body kinematics [39], can be written as follows:

$$\dot{\mathbf{r}}_i = \dot{\mathbf{r}} + \boldsymbol{\omega} \times \mathbf{d}_i, \tag{2.42}$$

$$\ddot{\mathbf{r}}_i = \ddot{\mathbf{r}} + \boldsymbol{\alpha} \times \mathbf{d}_i + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{d}_i), \tag{2.43}$$

where $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ are, respectively, the angular velocity and angular acceleration of the tray.
Considering the models described in section 2.4, the sloshing height of the i-th container can be expressed as follows:

damping coefficient. Therefore, the analyses conducted for the i-th container are valid in general for each container.

- for the L_p model:

$$\overline{\eta}_i = \frac{4hm_1}{m_F R}\sqrt{x_i^2 + y_i^2} + \frac{R^2\dot{\theta}^2}{4g}, \tag{2.44}$$

where the generalized coordinates x_i and y_i, referring to the first sloshing mode of the i-th container, are obtained by solving (2.38), provided below for convenience:

$$\begin{cases} \ddot{x}_i = -\omega_1^2 x_i - 2\omega_1\zeta_1\dot{x}_i - \ddot{r}_{i,x} - \ddot{r}_{i,z}\frac{C_1}{R}x_i \\ \\ \ddot{y}_i = -\omega_1^2 y_i - 2\omega_1\zeta_1\dot{y}_i - \ddot{r}_{i,y} - \ddot{r}_{i,z}\frac{C_1}{R}y_i \end{cases} \tag{2.45}$$

- for the NL model:

$$\overline{\eta}_i = \frac{\xi_{11}^2 hm_1}{m_F R}\sqrt{x_i^2 + y_i^2}, \tag{2.46}$$

where the generalized coordinates x_i and y_i, referring to the first sloshing mode of the i-th container, are the solution of the EOMs (2.20), provided below for convenience:

$$\begin{cases} \left(1 + \frac{C_1^2}{R^2}x_i^2\right)\ddot{x}_i + \frac{C_1^2}{R^2}x_i y_i \ddot{y}_i = -\frac{C_1^2}{R^2}(\dot{x}_i^2 + \dot{y}_i^2)x_i + (2\dot{\theta}\dot{y}_i + \dot{\theta}^2 x_i + \ddot{\theta}y_i) \\ \qquad -\omega_1^2 x_i\left[1 + \alpha_1(x_i^2 + y_i^2)^{w-1}\right] - 2\omega_1\zeta_1[\dot{x}_i + \frac{C_1^2}{R^2}(x_i\dot{x}_i + y_i\dot{y}_i)x_i] \\ \qquad -\ddot{r}_{i,x}\cos\theta - \ddot{r}_{i,y}\sin\theta - \ddot{r}_{i,z}\frac{C_1}{R}x_i \\ \\ \left(1 + \frac{C_1^2}{R^2}y_i^2\right)\ddot{y}_i + \frac{C_1^2}{R^2}x_i y_i \ddot{x}_i = -\frac{C_1^2}{R^2}(\dot{x}_i^2 + \dot{y}_i^2)y_i + (-2\dot{\theta}\dot{x}_i + \dot{\theta}^2 y_i - \ddot{\theta}x_i) \\ \qquad -\omega_1^2 y_i\left[1 + \alpha_1(x_i^2 + y_i^2)^{w-1}\right] - 2\omega_1\zeta_1[\dot{y}_i + \frac{C_1^2}{R^2}(x_i\dot{x}_i + y_i\dot{y}_i)y_i] \\ \qquad +\ddot{r}_{i,x}\sin\theta - \ddot{r}_{i,y}\cos\theta - \ddot{r}_{i,z}\frac{C_1}{R}y_i \end{cases} \tag{2.47}$$

A key aspect of the proposed optimizations is determining the time-optimal trajectory while ensuring that the sloshing height of each container remains within a specified limit for the entire duration of the trajectory. The sloshing-height limit is therefore a constraint in the optimization process and must be carefully formulated.
Let $\overline{\eta}_{lim}$ be the imposed sloshing-height limit; the following inequality must be satisfied:

$$\overline{\eta}_i \leq \overline{\eta}_{lim}. \tag{2.48}$$

By employing the formulation expressing $\overline{\eta}_i$, i.e. equations (2.44) and (2.46), we obtain equation (2.49) for the L_p model, and equation (2.50) for the NL model:

$$\frac{4hm_1}{m_F R}\sqrt{x_i^2 + y_i^2} + \frac{R^2\dot{\theta}^2}{4g} \leq \overline{\eta}_{lim}, \tag{2.49}$$

$$\frac{\xi_{11}^2 hm_1}{m_F R}\sqrt{x_i^2 + y_i^2} \leq \overline{\eta}_{lim}, \tag{2.50}$$

which can be elaborated to attribute the sloshing constraint to the generalized coordinates x_i, y_i:

$$\sqrt{x_i^2 + y_i^2} \leq \frac{m_F R}{4hm_1}\left(\overline{\eta}_{lim} - \frac{R^2\dot{\theta}^2}{4g}\right), \tag{2.51}$$

$$\sqrt{x_i^2 + y_i^2} \leq \frac{m_F R}{\xi_{11}^2 h m_1} \overline{\eta}_{lim}, \tag{2.52}$$

where (2.51) refers to the L_p model and (2.52) refers to the NL model.
In conclusion, in this section, a chain of dependencies among the involved variables has been highlighted. In fact, equations (2.51) and (2.52) link $\overline{\eta}_{lim}$ (which represents a constraint of the problem) to the generalized coordinates x_i and y_i, which are functions of the i-th container acceleration from equations (2.45) and (2.47). The acceleration of the i-th container is a function of the motion imposed on the tray, namely $\mathbf{r}(t)$ and $\theta(t)$, as provided by equations (2.41), (2.42), and (2.43). Therefore, starting from the imposed tray motion, the i-th sloshing height can be determined; consequently, given a constraint on the sloshing height, it is possible to design a motion law that satisfies such a constraint, which is precisely the objective of the optimizations presented in this thesis.

Chapter 3

Time-optimal trajectory planning for assigned paths

This chapter provides a detailed explanation of the developed algorithms for trajectories with a predefined path. The parameters that influence the optimization, including the control inputs, the initial input solution, and the number of constrained containers, are analyzed.
The aim of this chapter is to identify the most suitable strategy for performing the simulations in order to obtain the best possible solution, in terms of computational time and optimization efficacy; for this reason, in this exploration phase, the kinematics of the robot responsible for handling the containers is not considered. It will be addressed in chapter 5, in which the trajectory-planning algorithms for experimental tests will be developed, and maximum joint velocity limits will constitute an additional constraint for the optimization.

3.1 Trajectory definition

For the problem at hand, the path that the containers must follow is predefined; therefore, the objective of the optimization problem is to determine the optimal motion law that allows the execution of the trajectory in minimal time, while at the same time binding the sloshing height of the constrained containers under a specified threshold. For practical reasons, the path is parameterized in terms of a path parameter s, whose motion law $s(t), \dot{s}(t), \ddot{s}(t)$ describes the temporal evolution of the path, i.e, the time trajectory.
The vector $\mathbf{r}$ expressing the position of point O' can be written as:

$$\mathbf{r} = \mathbf{r}(s), \qquad s \in [0,1]. \tag{3.1}$$

A graphical representation of $\mathbf{r}(s)$ is shown in figure 3.1. Given the motion law $s(t)$ (to be sought by the optimization), the trajectory can be defined as follows:

$$\dot{\mathbf{r}}(s,\dot{s}) = \mathbf{r}'(s)\dot{s}, \tag{3.2}$$

$$\ddot{\mathbf{r}}(s,\dot{s},\ddot{s}) = \mathbf{r}''(s)\dot{s}^2 + \mathbf{r}'(s)\ddot{s}, \tag{3.3}$$

where the symbology $()' = \partial()/\partial s$ represents the derivative w.r.t. s. The relationship between $\mathbf{r}$ and s has been defined using B-spline basis function (see [40], [41]). These functions are particularly suitable and advantageous, for the proposed applications, as

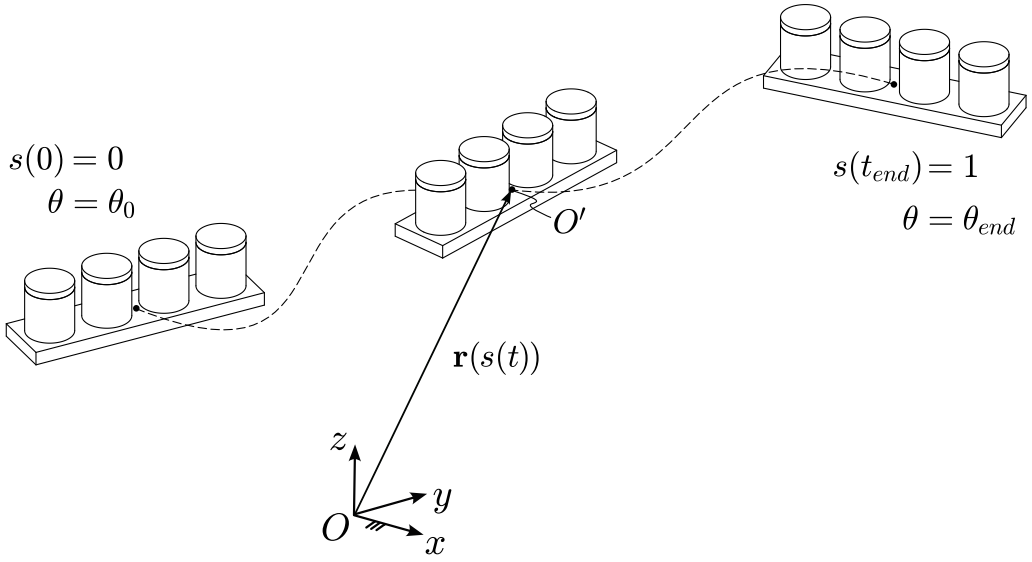


Figure 3.1: Schematic of the prescribed path.

they allow the definition of the 3D path as a function of s by simply specifying *control points*, i.e., points in space around which the path is intended to pass. Through B-spline functions, these points are connected, defining a path already parameterized as a function of s. The definition of $\mathbf{r}(s)$ using these functions is provided as follows:

$$\mathbf{r}(s) = \sum_{j=0}^{m} B_j^d(s)\mathbf{p}_j, \qquad s \in [0,1], \tag{3.4}$$

where $B_j^d(s)$ are the B-spline basis functions of degree d, in our case $d = 4$, and $\mathbf{p}_j$ represent the $m+1$ control points.
From equation (3.3), and recalling equation (2.43), the acceleration of each container can be determined as a function of s and θ by means of the following equation:

$$\ddot{\mathbf{r}}_i(s,\dot{s},\ddot{s},\theta,\dot{\theta},\ddot{\theta}) = \ddot{\mathbf{r}}(s,\dot{s},\ddot{s}) + \boldsymbol{\alpha}(\theta,\dot{\theta},\ddot{\theta}) \times \mathbf{d}_i(\theta) + \boldsymbol{\omega}(\theta,\dot{\theta}) \times \big(\boldsymbol{\omega}(\theta,\dot{\theta}) \times \mathbf{d}_i(\theta)\big), \tag{3.5}$$

where $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$ are, respectively, the angular velocity and angular acceleration of the container tray, and $\mathbf{d}_i$ represents the vector $(O_i - O')$ (figure 2.5) w.r.t. the fixed reference frame. From equation (3.5), it can be observed that the trajectory of each container, $\mathbf{r}_i(t)$, is uniquely defined by the functions $s(t)$ and $\theta(t)$.

3.2 Control input

Considering a system, described by the state vector $\mathbf{x}(t)$, the objective of time-optimal trajectory planning is to find the optimal control input $\mathbf{u}(t)$ that enables the trajectory to be executed in minimum time (by minimizing a cost functional), while satisfying all constraints imposed on both $\mathbf{u}(t)$ and the system state $\mathbf{x}(t)$ (see [42,43]). In this section, an investigation is conducted regarding which variables to use as control inputs for the problem and their influence on the final solution.
As discussed in chapter 2, the trajectory followed by each container, denoted by $\mathbf{r}_i(t)$, depends on two quantities, s and θ, so that $\mathbf{r}_i(t) = \mathbf{r}_i(s(t),\theta(t))$. By determining these two functions, it is possible to completely describe the system.

The chosen control input is:

$$\mathbf{u} = \begin{bmatrix} u_s \\ u_\theta \end{bmatrix} = \begin{bmatrix} \dddot{s} \\ \dddot{\theta} \end{bmatrix}. \tag{3.6}$$

The jerk of the two variables is chosen as control input and it is included in the cost functional; this way, minimizing the jerk ensures a smooth trajectory [40].
Alternatively, one can only select the jerk of the path parameter as control input, namely:

$$u = \dddot{s}. \tag{3.7}$$

hence defining θ as a linear function of s, i.e.:

$$\theta = \theta_0 + s \cdot (\theta_{end} - \theta_0), \qquad s \in [0,1], \tag{3.8}$$

where θ_0 and θ_{end} denote the values (which are determined *a priori*) at the beginning and at the end of the trajectory, respectively.

3.2.1 Problem Formulation with $\mathbf{u} = [\,\dddot{s}, \dddot{\theta}\,]^T$

This case refers to the control input defined in (3.6). Considering n_c containers, the system state is defined by a vector $\mathbf{x} \in \mathbb{R}^{6+4n_c}$, namely:

$$\mathbf{x} = [s\ \dot{s}\ \ddot{s}\ \theta\ \dot{\theta}\ \ddot{\theta}\ x_1\ y_1\ \dot{x}_1\ \dot{y}_1\ \ldots\ x_{n_c}\ y_{n_c}\ \dot{x}_{n_c}\ \dot{y}_{n_c}]^T. \tag{3.9}$$

The optimization problem can be formulated as:

$$\min_{t_{end},\mathbf{u}} \left[\int_0^{t_{end}} (1 + k_s u_s^2 + k_\theta u_\theta^2)\, dt \right] \tag{3.10a}$$

subject to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{3.10b}$$
$$\mathbf{x}(0) = [0\ 0\ 0\quad \theta_0\quad 0\ 0\ 0\ 0\ 0\ 0\ \ldots\ 0\ 0\ 0\ 0]^T \tag{3.10c}$$
$$\mathbf{x}(t_{end}) = [1\ 0\ 0\ \theta_{end}\ 0\ 0\ 0\ 0\ -\ -\ \ldots\ 0\ 0\ -\ -]^T \tag{3.10d}$$
$$\overline{\eta}_i(t) \le \overline{\eta}_{lim} \qquad i = 1,\ldots,n_c;\quad t \in [0, t_{end}] \tag{3.10e}$$
$$\overline{\eta}_i(t) \le 0.2\overline{\eta}_{lim} \qquad i = 1,\ldots,n_c;\quad t > t_{end} \tag{3.10f}$$
$$|u_s| \le u_{s,max};\quad |u_\theta| \le u_{\theta,max}. \tag{3.10g}$$

The cost functional in (3.10a) represents a trade-off between minimizing the trajectory duration and reducing jerk. The jerk terms are weighted by two constants, k_s and k_θ, which must be properly tuned (as k decreases, t_{end} decreases, but maximum accelerations increase). Typically, these constants are chosen within the range: $k_{s,\theta} \in [10^{-5}, 10^{-2}]$.
The function $\mathbf{f}$ in (3.10b) includes the integration chain of s and θ from $\mathbf{u}$ (3.11a) and

the sloshing dynamics of each container (3.11b), i.e.:

$$\frac{d}{dt}\begin{bmatrix} s \\ \dot{s} \\ \ddot{s} \\ \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{s} \\ \ddot{s} \\ u_s \\ \dot{\theta} \\ \ddot{\theta} \\ u_\theta \end{bmatrix}, \tag{3.11a}$$

$$\frac{d}{dt}\begin{bmatrix} x_i \\ y_i \\ \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ -2\zeta_1\omega_1\dot{x}_i - \omega_1^2 x_i - \ddot{r}_{i,x} - \left(\frac{\omega_1^2}{g}x_i\right)\ddot{r}_{i,z} \\ -2\zeta_1\omega_1\dot{y}_i - \omega_1^2 y_i - \ddot{r}_{i,y} - \left(\frac{\omega_1^2}{g}y_i\right)\ddot{r}_{i,z} \end{bmatrix} \qquad i = 1,...,n_c. \tag{3.11b}$$

The constraints in (3.10c, 3.10d) define the initial and final states of $\mathbf{x}$. In (3.10d), it can be observed that the symbols "-" appear in correspondence of $\dot{x}_i(t_{end})$ and $\dot{y}_i(t_{end})$ ($i = 1,...,n_c$), indicating that no final condition has been imposed on these variables. The constraint ensuring that the sloshing height of each container remains below $\overline{\eta}_{lim}$ is represented by the inequality constraint in (3.10e), while (3.10f) limits the residual liquid oscillations to $0.2\overline{\eta}_{lim}$ after the trajectory time t_{end}.

3.2.2 Problem Formulation with $u = \dddot{s}$ and $\theta = \theta(s)$

This case refers to the control input of equation (3.7).Considering n_c containers, the system state is defined by a vector $\mathbf{x} \in \mathbb{R}^{3+4n_c}$ (smaller by three dimensions compared to the previous case), namely:

$$\mathbf{x} = [s\ \dot{s}\ \ddot{s}\ x_1\ y_1\ \dot{x}_1\ \dot{y}_1\ ...\ x_{n_c}\ y_{n_c}\ \dot{x}_{n_c}\ \dot{y}_{n_c}]^T. \tag{3.12}$$

The optimization problem can be formulated as:

$$\min_{t_{end},u}\left[\int_0^{t_{end}}(1+ku^2)dt\right] \tag{3.13a}$$

subject to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},u) \tag{3.13b}$$

$$\mathbf{x}(0) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ ...\ 0\ 0\ 0\ 0]^T \tag{3.13c}$$

$$\mathbf{x}(t_{end}) = [1\ 0\ 0\ 0\ 0\ -\ -\ ...\ 0\ 0\ -\ -]^T \tag{3.13d}$$

$$\overline{\eta}_i(t) \le \overline{\eta}_{lim} \qquad i = 1,...,n_c;\ \ t \in [0, t_{end}] \tag{3.13e}$$

$$\overline{\eta}_i(t) \le 0.2\overline{\eta}_{lim} \qquad i = 1,...,n_c;\ \ t > t_{end} \tag{3.13f}$$

$$|u| \le u_{max}. \tag{3.13g}$$

The same considerations from section 3.2.1 can be applied regarding the cost functional (3.13a) and the formulated constraints (3.13c-3.13g). For completeness, the equa-

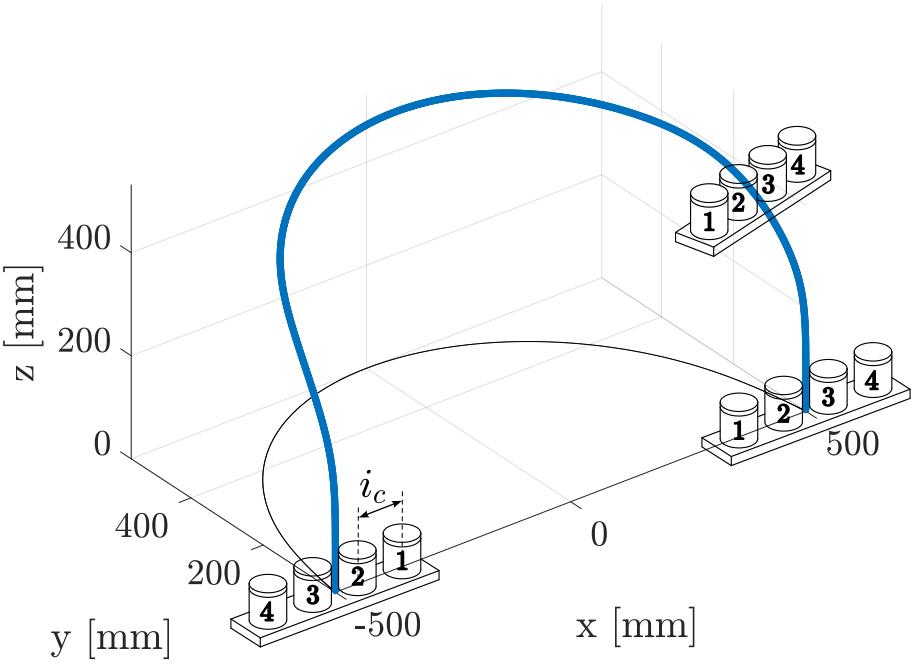


Figure 3.2: Considered path for control-input comparison.

tions of the integration chain represented by the function **f** in (3.13b) are provided:

$$\frac{d}{dt}\begin{bmatrix} s \\ \dot{s} \\ \ddot{s} \end{bmatrix} = \begin{bmatrix} \dot{s} \\ \ddot{s} \\ u \end{bmatrix}, \tag{3.14a}$$

$$\frac{d}{dt}\begin{bmatrix} x_i \\ y_i \\ \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ -2\zeta_1\omega_1\dot{x}_i - \omega_1^2 x_i - \ddot{r}_{i,x} - \left(\frac{\omega_1^2}{g}x_i\right)\ddot{r}_{i,z} \\ -2\zeta_1\omega_1\dot{y}_i - \omega_1^2 y_i - \ddot{r}_{i,y} - \left(\frac{\omega_1^2}{g}y_i\right)\ddot{r}_{i,z} \end{bmatrix} \qquad i = 1, \ldots, n_c. \tag{3.14b}$$

3.2.3 Comparison between different control inputs

The two control-input choices, presented in sections 3.2.1 and 3.2.2, are analyzed by comparing the corresponding optimization results, in terms of trajectory duration (obtained from the algorithm) and computational time. The prescribed path is shown in figure 3.2 and the following parameters are considered:

- container radius: $R = 49$ mm;
- liquid static height: $h = 80$ mm;
- number of containers: $n_c = 4$;
- containers center-to-center distance: $i_c = 120$ mm;
- cost-functional constants: $k_s = 10^{-3}[\mathrm{s}^6]$, $k_\theta = 10^{-4}[\mathrm{s}^6/\mathrm{rad}^2]$.

Table 3.1 shows the results of the optimizations with the two different control inputs, and by imposing different sloshing-height limits $\overline{\eta}_{lim}$. It can be noted that, for the same

value of $\overline{\eta}_{lim}$, the strategy with $\mathbf{u} = [\dddot{s}, \ddot{\theta}]$, despite increasing the algorithm complexity, only provides a minimal reduction in trajectory duration. Therefore, from this point onward, reference will be made to the optimization model described in section 3.2.2, where the control input is represented by $u = \dddot{s}$ and θ is a linear function of s (see (3.8)). Moreover, by analyzing the computation times t_{calc}, it can be observed that in some cases they even double when using $\mathbf{u} = [\dddot{s}, \ddot{\theta}]$ instead of $u = \dddot{s}$. For this reason, since a very large number of optimizations were performed in the subsequent analyses, $u = \dddot{s}$ was used also to accelerate the process.

Table 3.1: Comparison between control inputs.

$\overline{\eta}_{lim}$	$\mathbf{u} = [\dddot{s}, \ddot{\theta}]$	$u = \dddot{s}$
10 mm	$t_{end} = 2.82s$	$t_{end} = 3.07s$
	$t_{calc} = 244.60s$	$t_{calc} = 125.83s$
20 mm	$t_{end} = 2.29s$	$t_{end} = 2.45s$
	$t_{calc} = 149.15s$	$t_{calc} = 86.52s$
30 mm	$t_{end} = 2.08s$	$t_{end} = 2.18s$
	$t_{calc} = 76.68s$	$t_{calc} = 104.04s$

3.3 Considerations on local/global minima

Regarding the resolution of the optimization algorithm, an aspect that has to be taken into account is that, depending on the chosen initial-guess motion law s_{input}, the achievement of a global solution is not granted *a priori* (a detailed analysis of this aspect is provided in section 3.3.1). Since the main objective of this thesis is to provide fast and efficient methods for the time-optimal trajectory planning of high-complexity problems, the thorough analysis of globalization techniques lies beyond the scope of the presented work. It is worth mentioning that the literature offers various techniques for obtaining global minima, such as multi-start algorithms [44] or particle swarm optimization [31]. In particular, in the latter case, a computational time of 1006.1 s is reported in the determination of 5 optimal values. In our case, the problem at hand is more complex, and this can be attributed to:

- the large-dimension arrays involved, due to the fact that the continuum time domain has to be discretized in a sufficient number of time steps;
- the EOMs reproducing the liquid sloshing behavior introduce strong non-linearities inside the dynamical system that is incorporated within the optimization algorithm.

Notwithstanding the aforementioned limitations, the proposed resolution algorithms offer acceptable solutions within reasonable computational times, of just tens of seconds. This comparison underscores the advantage of our approach: despite dealing

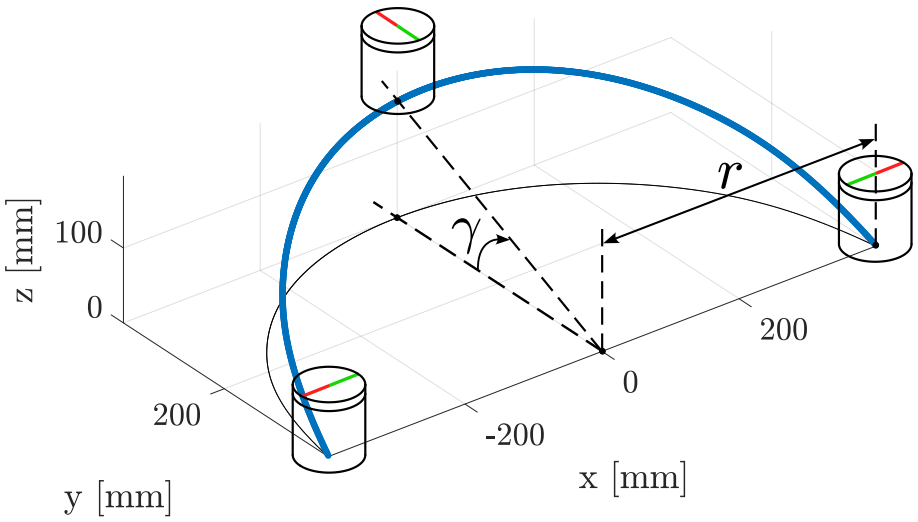


Figure 3.3: Trajectory parameters.

with a problem of much higher dimension and more complex constraints, our method achieves convergence in a fraction of the time.

3.3.1 Initial attempt trend of s

Since a global solution is not obtained, it is necessary to identify the parameters for which the optimizer converges to a local minimum or another. In order to initiate the optimization process, the optimizer must be provided with an initial trend of s, denoted as s_{input}, from which it will iteratively converge to the final solution. Therefore, the trend s_{input} may influence the obtained results. In this section, we analyze the effect of s_{input} on the final solution. Numerous simulations have been performed for this purpose, considering the following scenario:

- a single container, for simplicity, centered at point O' (figure 2.5);
- the semicircular path in figure 3.3, characterized by a radius r and an inclination angle with respect to the horizontal plane equal to γ;

by varying the following parameters:

- the container radius R;
- the imposed sloshing-height limit $\overline{\eta}_{lim}$;
- the path radius r (figure 3.3);
- the angle γ (figure 3.3).

For each parameter set, an optimization was performed by providing six different s_{input} trends to the algorithm, to assess whether varying this trend could lead to a different solution. Should this occur, the objective was to evaluate which specific s_{input} most frequently resulted in a better solution, i.e., a solution characterized by a smaller t_{end}. For the initial guess s_{input}, the ratio is to choose, among the ones available, the motion law that satisfies the following conditions: the path parameter s needs to start from 0 and to increase monotonically to 1. Based on the most common motion laws, different candidate initial guesses s_{input} of the algorithm can be considered: the ones used for our analysis are illustrated in figure 3.4. In particular:

- s(t) as a linear function, figure 3.4a;
- $\ddot{s}$(t) as a piecewise constant function, figure 3.4b;
- $\ddot{s}$(t) as a linear function, figure 3.4c;
- $\ddot{s}$(t) as a 5^{th}-degree polynomial function, figure 3.4d;
- $\ddot{s}$(t) as a 7^{th}-degree polynomial function, figure 3.4e;
- $\ddot{s}$(t) as a modified trapezoidal function, figure 3.4f.

Each s_{input} was chosen with a duration of 5 s, in any case, no significant differences were observed when varying this value around this setting.

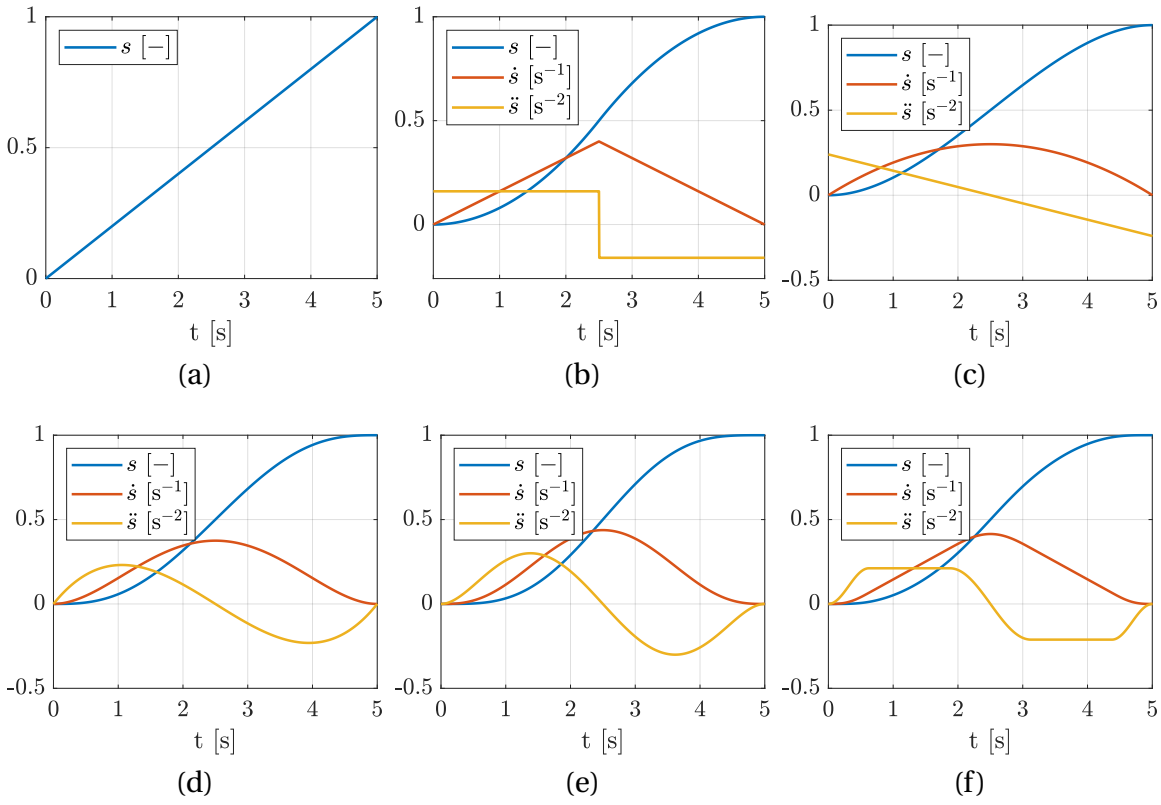


Figure 3.4: s_{input} trends.

For instance, the results obtained with the following parameter set are shown in figure 3.5: $R = 49$ mm, $r = 400$ mm, $\gamma = \pi/2$, and $\overline{\eta}_{lim} = 30$ mm. Specifically, the figure presents the trends of the optimized motion laws obtained starting from the six different s_{input} values, along with the trajectory duration t_{end} for each optimization. The corresponding s_{input} to each sub-figure in 3.5 is the counterpart in figure 3.4, meaning that, for example, the optimized trends in 3.5a are obtained by choosing the trends shown in 3.4a as s_{input}, and so on.
It can be observed that, depending on the choice of s_{input}, the optimizer may converge to a motion-law type or another: the first motion-law type has a duration of $t_{end} = 1.7239s$ and it is obtained with the s_{input} trends illustrated in figures 3.4a, 3.4c, 3.4d, 3.4f, whereas the second motion-law family is characterized by a $t_{end} = 1.5319s$

and it is obtained with the s_{input} trends of figures 3.4b, 3.4e. Therefore, depending on which s_{input} trend is used, the percentage difference in the obtained final time is equal to:

$$\Delta t_{end,\%} = 100 \frac{1.7239\ \text{s} - 1.5319\ \text{s}}{1.7239\ \text{s}} = 11.14\%. \tag{3.15}$$

Since it has been verified that the initial attempt trend influences the obtained final solution, we now aim to determine whether one of these trends generally yields a solution with the smaller t_{end} as the four parameters $(R, r, \gamma, \overline{\eta}_{lim})$ vary. To this end, for each s_{input}, the t_{end} values obtained for every parameter set were summed. A total of 77 different parameter sets were analyzed. Thus, if one s_{input} consistently produces a lower t_{end} compared to the others across most parameter sets, the sum of the 77 t_{end} values obtained with that particular s_{input} should be significantly lower than the sums corresponding to the other s_{input} trends. The sums of the 77 times obtained for each s_{input} are shown in table 3.2.

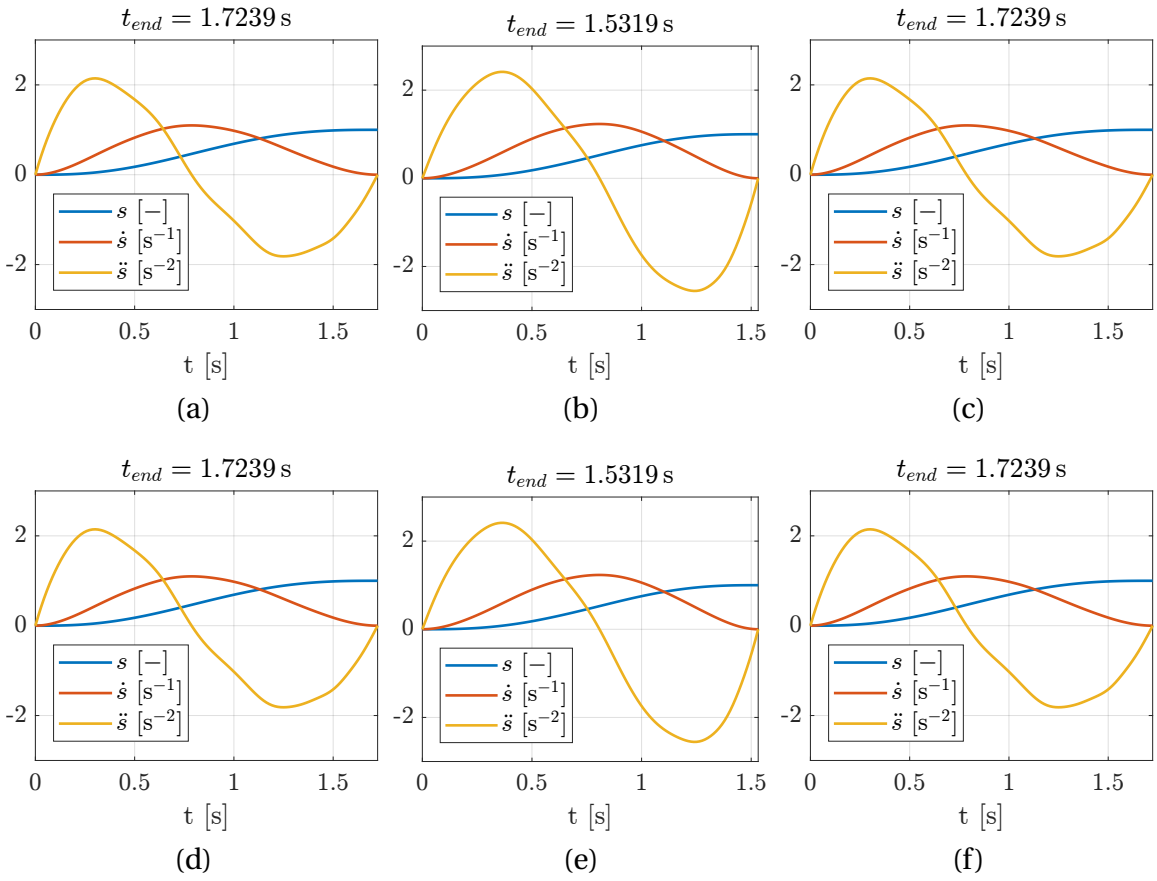


Figure 3.5: Optimized s trends.

Table 3.2: Total times with different s_{input}.

s_{input}	**Linear**	**Triangular**	**Cubic**	**Poly 5**	**Poly 7**	**Trapezoidal**
$t_{TOT} = \sum_{i=1}^{77} t_{end,i}$	133.45s	134.08s	134.66s	134.41s	134.52s	133.70s

By examining table 3.2, it can be observed that no total time is noticeably lower than

the others. In fact, although the difference between the t_{end} values obtained with different s_{input} trends can be around 10% for a single optimization, as shown by equation (3.15), this difference is reduced when considering all the optimization cases, dropping to 0.89%. This suggests that, among the candidate ones, there is no s_{input} trend that generally leads to a lower t_{end}.
Therefore, in conclusion, if the goal is to achieve the minimum possible time, it is advisable to perform all the tests with the different s_{input} trends for each individual case to identify the one that makes the algorithm converge to the minimal-time motion law. This multi-start exploration is feasible and can be performed in a reasonable amount time, since each simplified optimization (the robot kinematics is not considered and only a container is taken into account) runs in a very short computational time. In any case, if multiple solutions are obtained with different inputs, the difference is never significant.

3.3.2 Effectiveness of optimized motion law

To demonstrate the effectiveness of the optimized motion laws presented in the previous section compared to the non-optimized ones, this section presents a comparison of the sloshing-height trends obtained using the optimized motion law in figure 3.5b, with those obtained using standard motion laws of equal duration t_{end}. Specifically, the comparison includes a 7^{th}-degree polynomial motion law and a modified trapezoidal motion law. The resulting profiles for these three motion laws are shown in figure 3.6. It is important to recall that the optimization was performed by imposing a constraint of $\overline{\eta}_{lim} = 30$ mm.

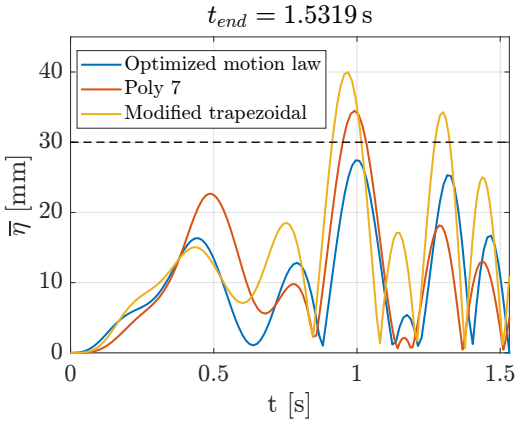
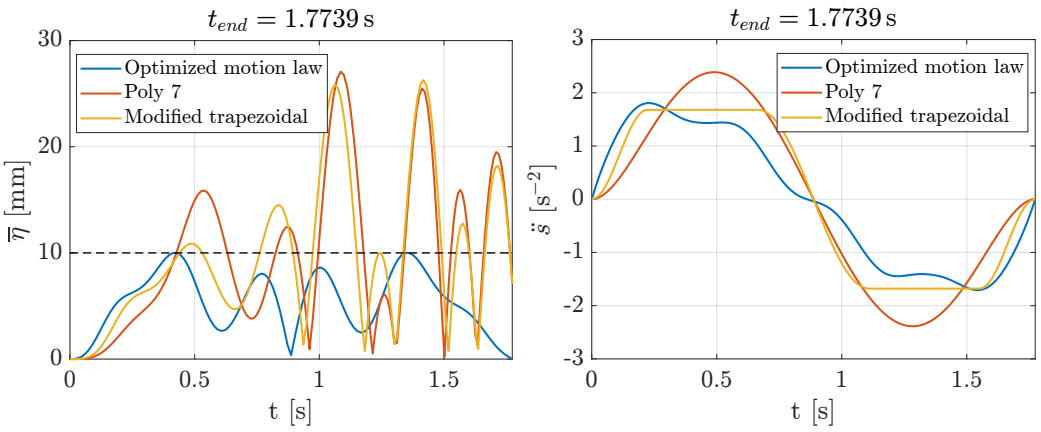


Figure 3.6: Comparison among sloshing-heights trends resulting from optimized and non-optimized motion laws.

It can be observed that non-optimized motion laws result in greater sloshing heights compared to the optimized one, exceeding the imposed limit $\overline{\eta}_{lim}$. The advantage of using the optimized motion law becomes more evident when reducing $\overline{\eta}_{lim}$. Figure 3.7 presents the results obtained by imposing $\overline{\eta}_{lim} = 10$ mm. Specifically, figure 3.7a shows the sloshing-height trends, while figure 3.7b shows the comparison between the motion laws (specifically, the accelerations $\ddot{s}$) used to obtain the sloshing-height trends. In this second case the benefit of employing the optimized motion law is evident. The difference between optimized and non-optimized motion laws becomes even more pronounced when considering the case of multiple containers, further highlighting the

necessity of optimization. Chapter 5 will present the results obtained from both simulations and experiments using optimized and non-optimized motion laws with multiple containers. In such cases, non-optimized motion laws lead to sloshing heights up to 70% higher than those obtained with optimized laws.



(a) Sloshing-height trends. (b) $\ddot{s}$ trends.

Figure 3.7: Comparison among sloshing heights and motion laws in optimized and non-optimized cases.

3.3.3 Remarks

Taking as a reference the results discussed in section 3.3.1, we provide below some considerations that further justify the choice not to adopt a global solution search strategy for these applications. We will highlight the complexity of the problems addressed and the resulting computational difficulties. In the graphs shown in figure 3.5, as previously explained, two possible solutions for the analyzed problem are represented. To better understand the nature of these solutions and the concept of a problem minimum, we consider figure 3.8. This figure represents the set of possible solutions for a 2-dimensional problem. Each point on the surface corresponds to a solution, defined by its two coordinates in the xy plane, while its height along the z axis represents the value of the functional F associated with that solution, i.e. the value pf $F(x,y)$. It can be observed that, as the chosen point on the surface varies, the functional value changes accordingly. The objective of our optimizations is to identify the point on the surface that has the lowest elevation along the z axis, i.e., the minimum of the functional. Drawing an analogy with figure 3.8, we can interpret the two solutions represented in figure 3.5 as two local minima of the surface in 3.8. From a visual perspective, it is evident that the choice of the starting point influences convergence towards one minimum rather than another: depending on the initial position on the surface, the optimization algorithm may lead to one valley or another. These starting points are nothing more than the s_{input} previously analyzed. The adoption of a multistart strategy for searching the global minimum consists of considering a certain number of s_{input} distributed over the entire surface. If these initial points are chosen with sufficient density, some of them will be close to the deepest valley, thereby allowing the identification of the global minimum. However, this strategy becomes very difficult when dealing with high-dimensional problems. The simplicity of the representation in figure 3.8, which refers to a 2-dimensional problem, is no longer applicable in our

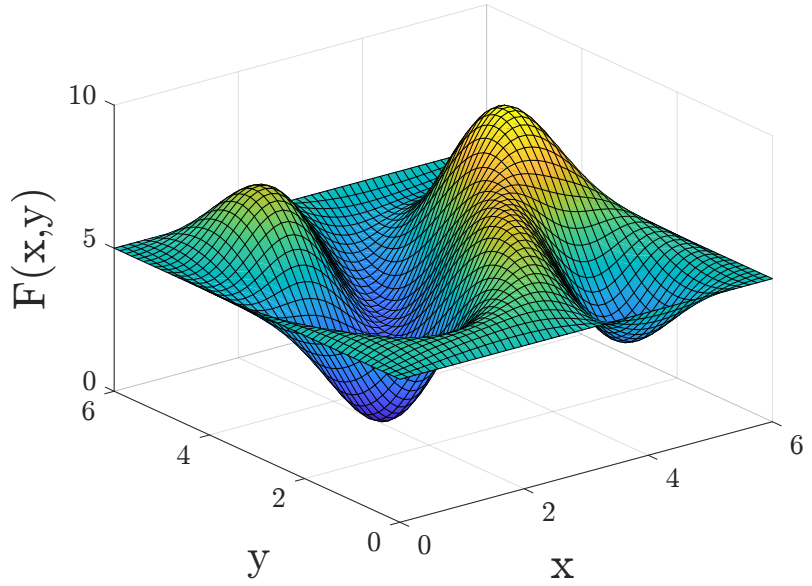


Figure 3.8: 2-dimensional minimization problem.

case. In fact, the problem under analysis has a dimensionality of 151 (the given s_{input} is an array of 151 components), far greater than 2. This means that each point on the surface is not characterized by only two coordinates, x and y, but rather by 151 components along 151 distinct axes. Consequently, applying a multistart algorithm, which would need to distribute the s_{input} in the space $\mathbb{R}^{151}$, becomes extremely difficult and expensive in computational terms. In conclusion, given the high complexity of the problem, determining the global minimum solution would require an enormous computational effort, making it infeasible for the purposes of the applications for which the developed algorithms are designed. For this reason, global optimization is not the objective of this thesis.

3.4 Influence of the container number

As the number of containers increases, the complexity of the problem also increases, as for each additional container, the system state $\mathbf{x}$ increases by 4 dimensions, and two inequality constraints are added, represented by inequalities (3.13e, 3.13f). However, it may not be necessary to consider all the containers *a priori* in the optimization. In this section, an analysis will be carried out to determine the minimum number of containers to include in the optimization, while still satisfying the maximum sloshing-height limit for each container. For this purpose, numerous simulations have been conducted considering different scenarios, in terms of:

- *the number of containers:* configurations with 4, 6, and 8 containers are considered, as shown in figure 3.9, which also includes the naming conventions used to refer to each container from now on: *IC* and *EC* denote the internal and external containers at the initial time of the trajectory, respectively; the remaining containers follow the numbering from $2, ..., (n_c - 1)$;

- *the trajectories:* considering the two paths $\mathbf{r}(s)$ shown in figures 3.10 and 3.11, the three considered trajectories are defined as follows:

– *Trajectory 1A*: it is inherited from Path 1 by setting $\theta_0 = 0$, $\theta_{end} = \pi$; as shown in figure 3.12 where, for the sake of clarity, the configuration with 4 containers is represented, the tray performs a total rotation equal to π.

– *Trajectory 1B*: it is inherited from Path 1 with $\theta_0 = 0$, $\theta_{end} = 2\pi$; as shown in figure 3.13 where, for the sake of clarity, the configuration with 4 containers is represented, the tray performs a total rotation equal to 2π.

– *Trajectory 2*: it is inherited from Path 2 with $\theta_0 = 0$, $\theta_{end} = \pi$; as shown in figure 3.14 where, for the sake of clarity, the configuration with 4 containers is represented, the tray performs a total rotation equal to π.

For each trajectory and for each container configuration, three optimization strategies will be investigated, namely:

- constraining the sloshing height on all containers;
- constraining the sloshing height on a single outer container (e.g., EC);
- constraining the sloshing height on both the outer containers (e.g., IC, EC).

The optimizations will be performed by imposing $\overline{\eta}_{lim} = 15$ mm, and considering a center-to-center distance between the containers of $i_c = 150$ mm.

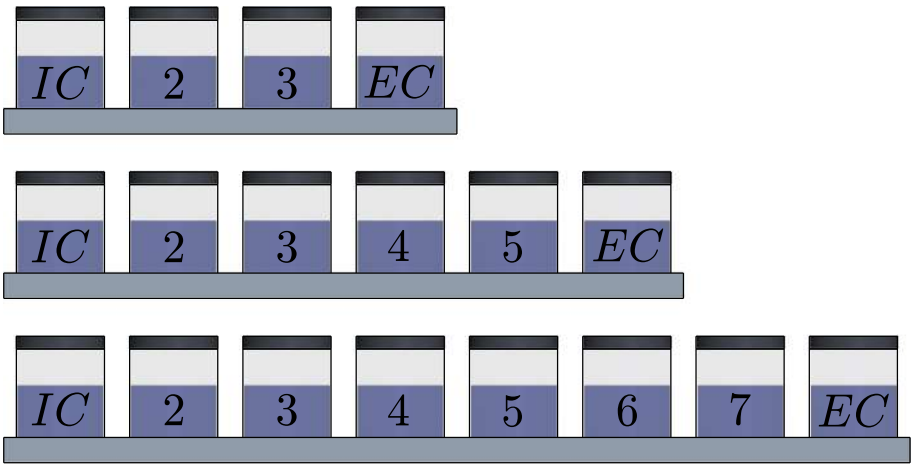


Figure 3.9: Container configurations.

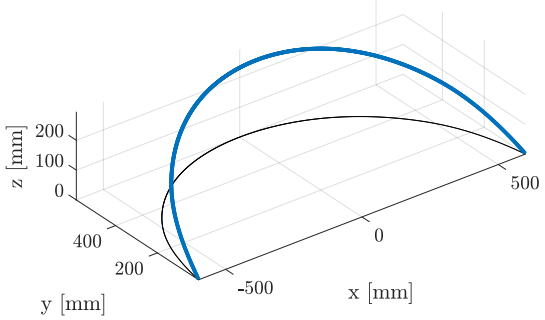


Figure 3.10: Path 1.

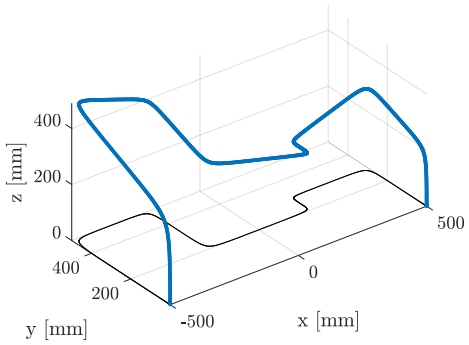


Figure 3.11: Path 2.

In the following figures, the results of the optimizations described in section 3.4 are

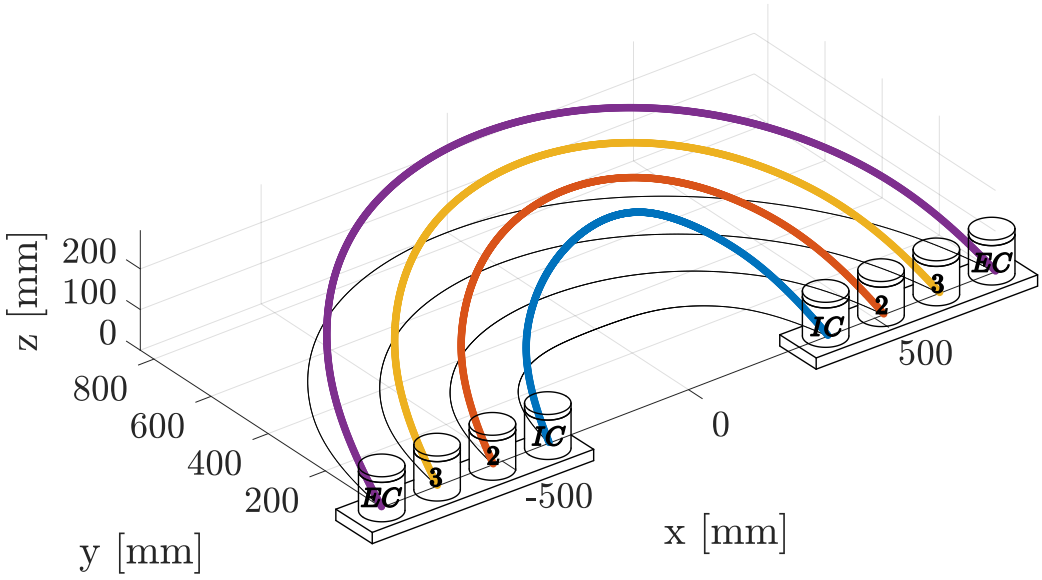


Figure 3.12: Trajectory 1A.

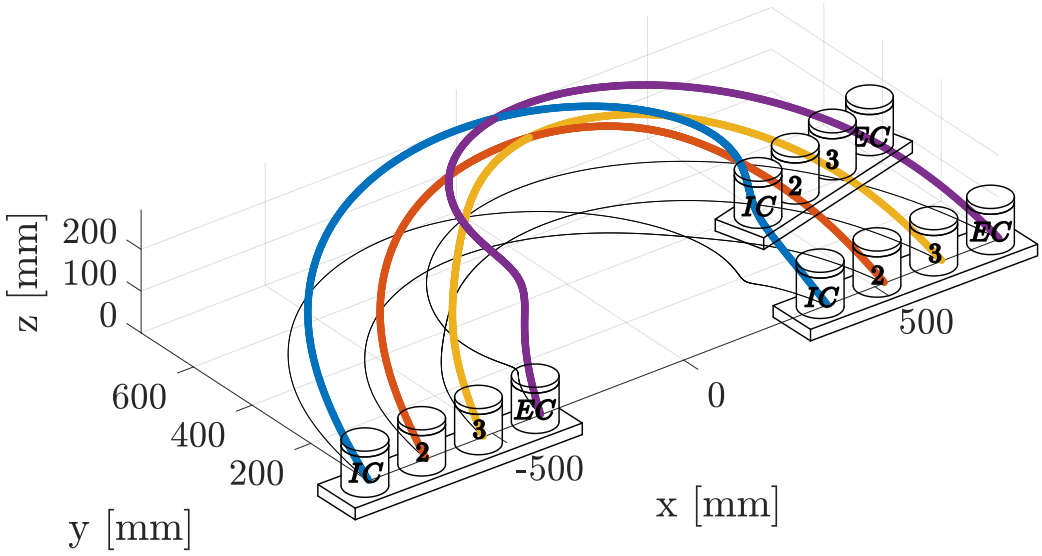


Figure 3.13: Trajectory 1B.

presented. Specifically, the trends of the sloshing heights for each container during the trajectories are shown. The results for *Trajectory 1A* are reported in figure 3.15, in particular in figure 3.15a the sloshing heights of all the containers have been constrained, in figure 3.15b the sloshing height of only *EC* has been constrained, and in figure 3.15c the sloshing heights of both *IC* and *EC* have been constrained. Similarly, the results are also reported for *Trajectory 1B* (figure 3.16) and *Trajectory 2* (figure 3.17). The mentioned figures present the results only for the configuration with 8 containers; the results for the configurations with 4 and 6 containers are reported in appendix A. For clarity, each figure caption specifies the specific case to which it refers, in terms of: trajectory, number of containers, and on which container the sloshing height has been constrained.

By analyzing the graphs, the following considerations can be made. First, it is observed that constraining all the sloshing heights always ensures compliance with the maximum limit for all the containers. On the other hand, constraining only the sloshing

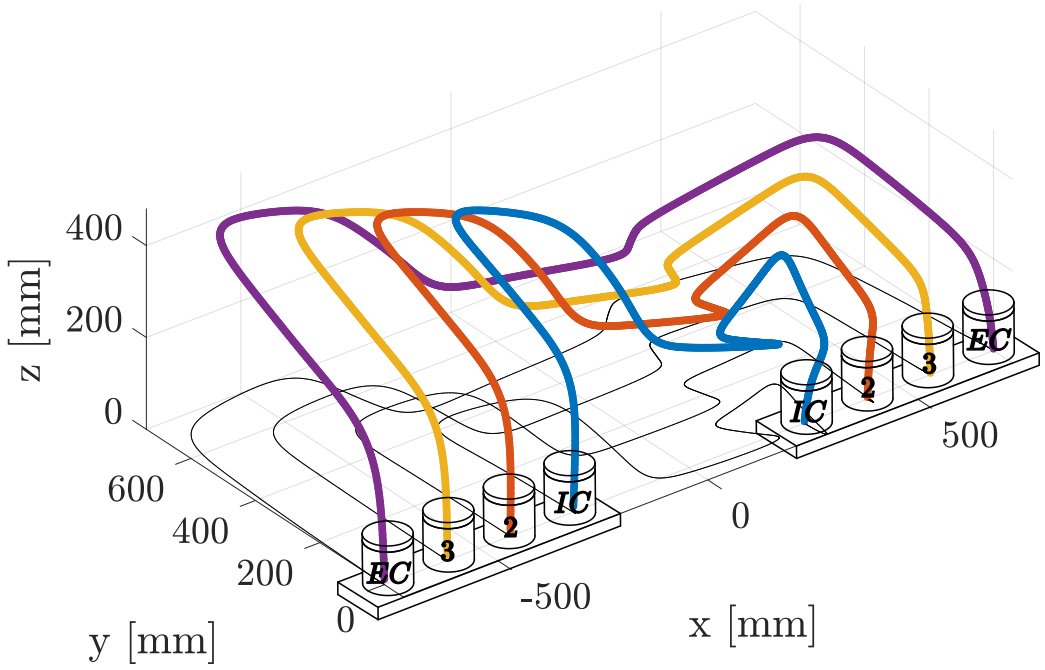


Figure 3.14: Trajectory 2.

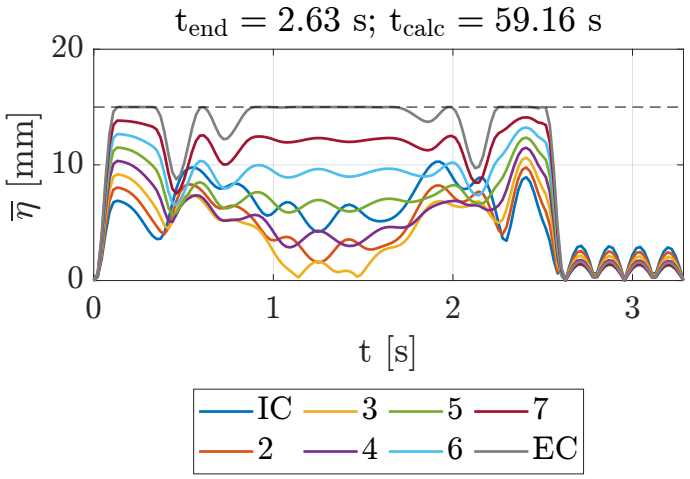
height of *EC* allows to satisfy these conditions for all the containers only if *EC* is subjected to the highest accelerations throughout the entire trajectory. This phenomenon occurs exclusively for Trajectory 1A (figures A.1b, A.2b, 3.15b), in which *EC* remains external throughout the entire trajectory (figure 3.12), thus experiencing the highest accelerations and representing the worst case for the specific trajectory to be optimized. However, for a generic trajectory, it is unlikely that a single container will be subjected to this condition throughout the entire motion, making this strategy less advantageous.

In contrast, constraining both internal and external containers proves beneficial because it ensures compliance with the sloshing-height limits for all containers while requiring less computational time compared to constraining all containers. The only exception to this statement is the case illustrated in figures 3.17a and 3.17c, where the optimization on all containers not only results in lower computational time but also leads to a slight reduction in the trajectory duration. This phenomenon can be attributed to the fact that, as discussed in section 3.3, the optimizer may converge to a local minimum. However, in most of the analyzed cases, the optimization constraining only the *IC* and *EC* sloshing heights proves to be the most efficient strategy, and even in the one scenario where this is not the case, the solution obtained is still very similar (t_{end} differs by only $0.01s$, which represents a negligible value), and the computational time remains limited.

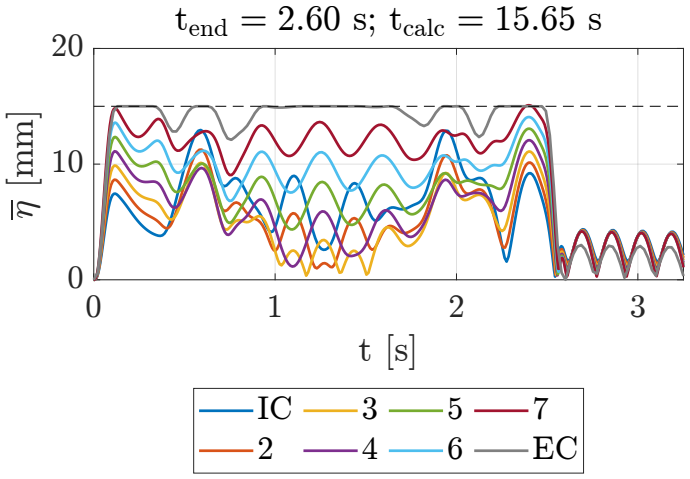
Another advantage of limiting the optimization on *IC* and *EC* is the scalability of the algorithm w.r.t. the number of containers. Indeed, as the number of containers increases, it is sufficient to increase the distance between *IC* and *EC* without the need to modify the size of the system state or introduce new constraint equations.

An additional consideration is that, although constraining only the *IC* and *EC* sloshing heights allows the sloshing limit to be satisfied for all containers, in some graphs this property appears not to be met. For example, in figure 3.16c, the limit is violated for

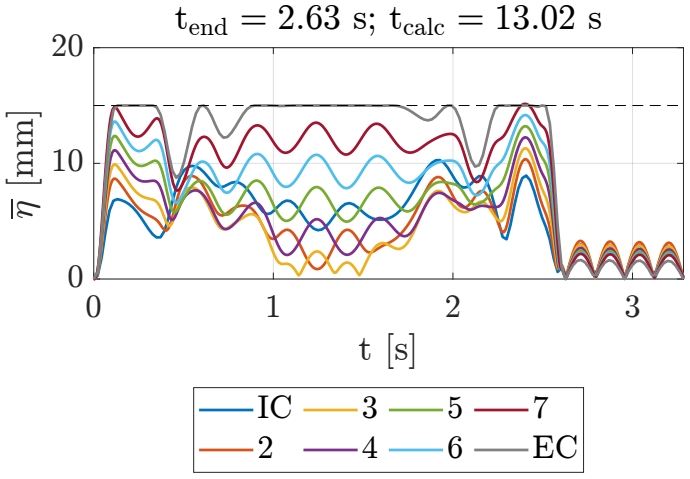
container 7. However, this is due solely to a computational approximation. Indeed, the curve shown is not a direct result of the optimizer but it is recalculated afterward based on the optimized profiles of *IC* and *EC*. These profiles are returned as arrays with 151 components, a relatively limited sampling rate. Consequently, when calculating the profiles of the other containers, which involves solving a second-order differential system, the solver applies interpolation to the available data, introducing an error that alters the profile trends. A confirmation of this statement can be found by comparing the curves in figure 3.16c with those in figure 3.16a: the values of t_{end} and the profiles of *IC* and *EC* are indeed exactly the same, thus representing the same solution, and indicating that the discrepancy in the sloshing-height profiles is due to numerical error in solving the differential equations. Since the solution is the same, the sloshing-height trend for each container must also be the same.



(a) $\overline{\eta}_{lim}$ on all.

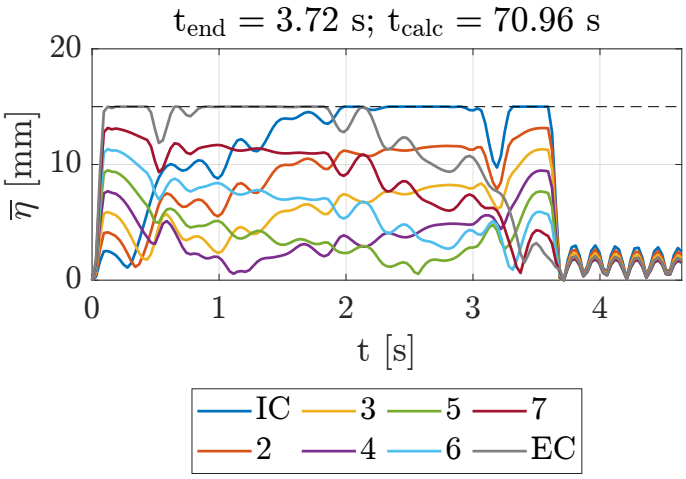


(b) $\overline{\eta}_{lim}$ on EC.

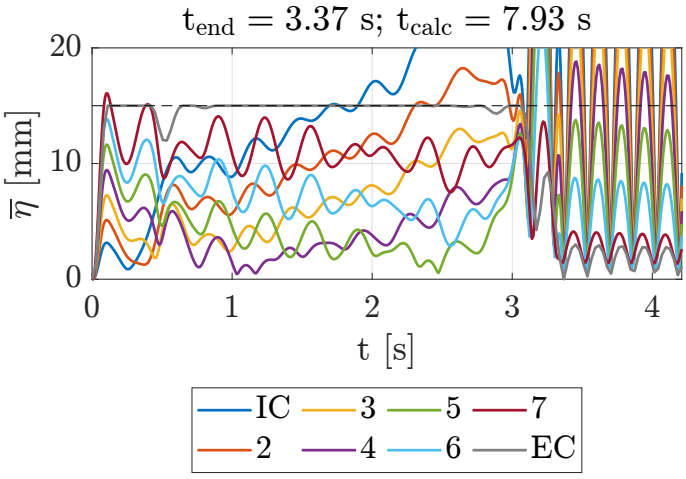


(c) $\overline{\eta}_{lim}$ on IC and EC.

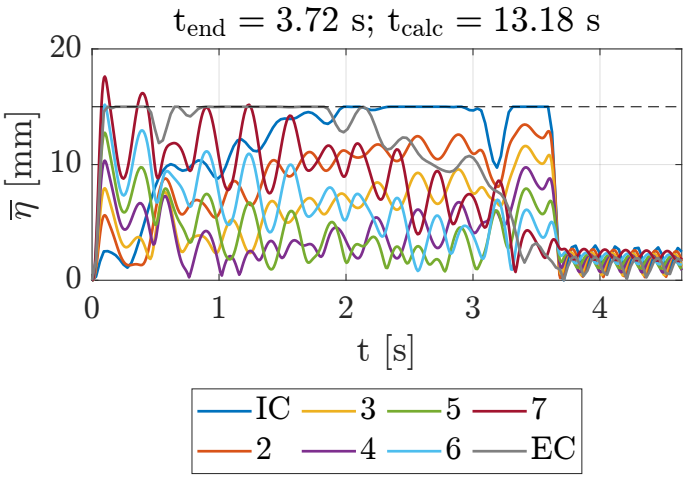
Figure 3.15: Trajectory 1A, 8 containers.



(a) $\overline{\eta}_{lim}$ on all.

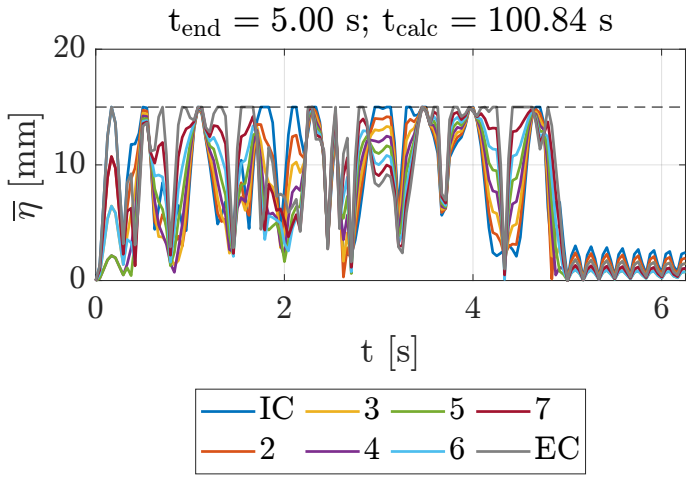


(b) $\overline{\eta}_{lim}$ on EC.

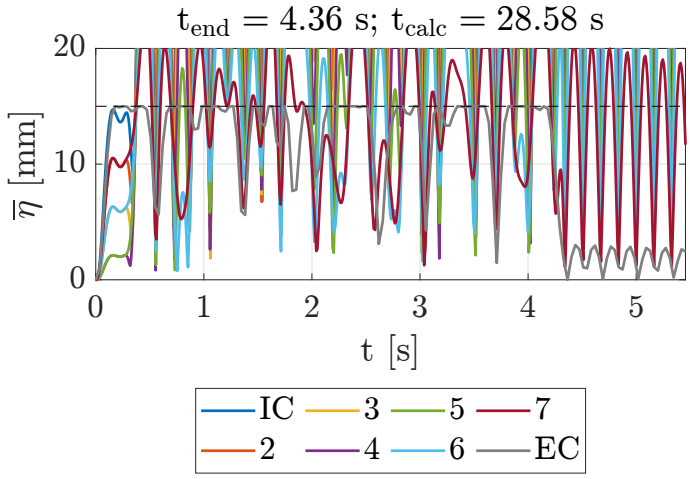


(c) $\overline{\eta}_{lim}$ on IC and EC.

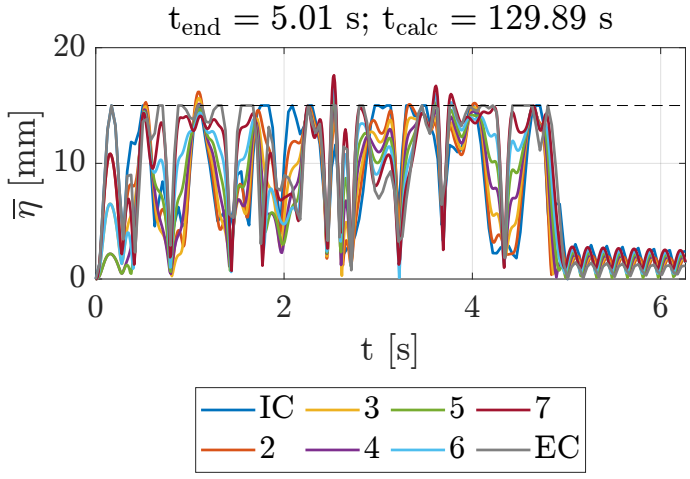
Figure 3.16: Trajectory 1B, 8 containers.



(a) $\overline{\eta}_{lim}$ on all.



(b) $\overline{\eta}_{lim}$ on EC.



(c) $\overline{\eta}_{lim}$ on IC and EC.

Figure 3.17: Trajectory 2, 8 containers.

Chapter 4

Time-optimal trajectory planning for point-to-point motion

Starting from the results of chapter 3, in this chapter, algorithms and results related to time-optimal trajectory planning along paths that are not assigned *a priori* will be analyzed. Only fixed points (section 4.2) or points whose position can vary within fixed volumes (section 4.3) will be assigned to the optimizer, and the task of the optimization will be to find the optimal path passing through these points, along with the optimal motion law. The first case, where the path must pass through the assigned points, will be referred to as *point-to-point motion with assigned waypoints*. The second case, where the points through which the path must pass can vary within assigned volumes, will be referred to as *point-to-point motion with assigned way-volumes*, and in this case, the optimal position of each point within its assigned volume will also be determined by the optimizer. As in the previous chapter, the robot kinematics is not included within the considered optimization algorithms.
The same container configurations and nomenclature used in figure 3.9 will be applied to these analyses. Based on the results presented in section 3.4, only the sloshing heights of IC and EC will be constrained. The paths from section 3.4 (figures 3.10 and 3.11), will be used as references to define the waypoints and way-volumes, allowing a meaningful comparison of results.

4.1 Control input

Unlike the models presented in sections 3.2.1 and 3.2.2, since no path is predefined, a path parameter s cannot be used. For this reason, the system state $\mathbf{x}$ directly includes the position array $\mathbf{r} \in \mathbb{R}^3$, which identifies the position of O' w.r.t. the fixed frame $Oxyz$ (figure 2.5). Given the absence of the parameter s, the angle θ is treated as an independent variable. Therefore, based on the same considerations made in section 3.2, the control input is defined as:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_r \\ u_\theta \end{bmatrix} = \begin{bmatrix} \dddot{\mathbf{r}} \\ \dddot{\theta} \end{bmatrix} = \begin{bmatrix} \dddot{r}_x \\ \dddot{r}_y \\ \dddot{r}_z \\ \dddot{\theta} \end{bmatrix}, \tag{4.1}$$

therefore, the optimizer will provide the optimal trends of $\mathbf{r}(t)$ and $\theta(t)$.

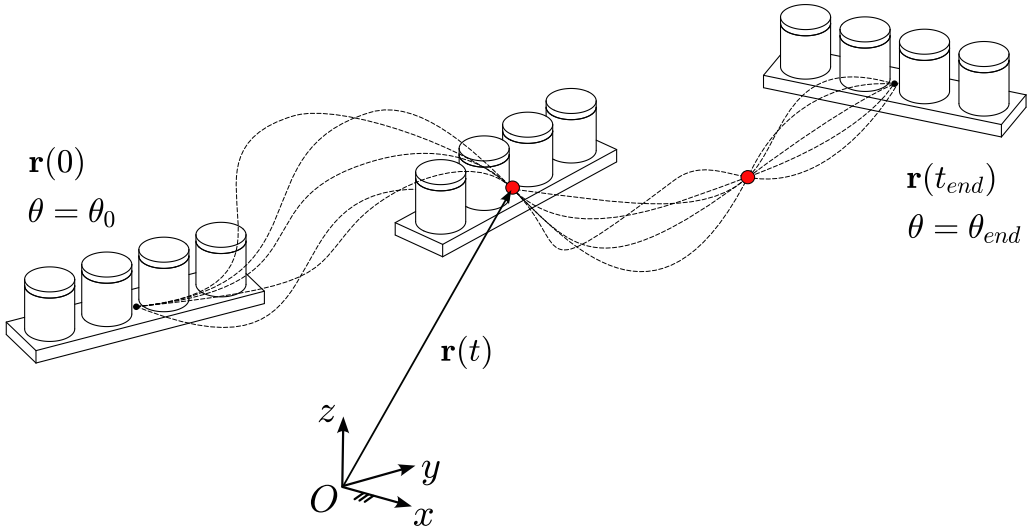


Figure 4.1: Point-to-point with assigned waypoints generic trajectories.

4.2 Assigned waypoints

Figure 4.1 shows a graphical representation of the problem at hand. It can be observed that multiple paths are possible, as in these optimizations only meaningful waypoints (highlighted in red in the figure) are assigned and the path is free.

4.2.1 Problem formulation

The state of the system is represented by the array $\mathbf{x} \in \mathbb{R}^{20}$, namely:

$$\mathbf{x} = [\mathbf{r}^T \ \dot{\mathbf{r}}^T \ \ddot{\mathbf{r}}^T \ \theta \ \dot{\theta} \ \ddot{\theta} \ x_{IC} \ y_{IC} \ \dot{x}_{IC} \ \dot{y}_{IC} \ x_{EC} \ y_{EC} \ \dot{x}_{EC} \ \dot{y}_{EC}]^T. \tag{4.2}$$

The optimization problem can be formulated as:

$$\min_{t_{end},\mathbf{u}} \left[\int_0^{t_{end}} (1 + k_r \mathbf{u}_r^T \mathbf{u}_r + k_\theta u_\theta^2) dt \right] \tag{4.3a}$$

subject to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{4.3b}$$
$$\mathbf{x}(0) = [\ \mathbf{r}_0^T \ \ \mathbf{0}^T \ \mathbf{0}^T \ \ \theta_0 \ \ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \tag{4.3c}$$
$$\mathbf{x}(t_{end}) = [\mathbf{r}_{end}^T \ \mathbf{0}^T \ \mathbf{0}^T \ \theta_{end} \ 0\ 0\ 0\ 0 - - 0\ 0 - -]^T \tag{4.3d}$$
$$\overline{\eta}_i(t) \leq \overline{\eta}_{lim} \qquad i = IC, EC; \ \ t \in [0, t_{end}] \tag{4.3e}$$
$$\overline{\eta}_i(t) \leq 0.2\overline{\eta}_{lim} \qquad i = IC, EC; \ \ t > t_{end} \tag{4.3f}$$
$$|\mathbf{u}_r| \leq \mathbf{u}_{r,max}; \ \ |u_\theta| \leq u_{\theta,max} \tag{4.3g}$$
$$\mathbf{r}(t_j) = \mathbf{r}_j \qquad j = 1, ..., n_p; \ \ t_j \in (t_0, t_{end}) \tag{4.3h}$$
$$r_z \in [r_{z,min}, r_{z,max}]. \tag{4.3i}$$

The function **f** in (4.3b) includes the integration chain of **r** and θ from **u** (4.4a) and the sloshing dynamics of each container (4.4b), namely:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{r}\\ \dot{\mathbf{r}}\\ \ddot{\mathbf{r}}\\ \theta\\ \dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{r}}\\ \ddot{\mathbf{r}}\\ \mathbf{u}_r\\ \dot{\theta}\\ \ddot{\theta}\\ u_\theta\end{bmatrix}, \tag{4.4a}$$

$$\frac{d}{dt}\begin{bmatrix}x_i\\ y_i\\ \dot{x}_i\\ \dot{y}_i\end{bmatrix} = \begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ -2\zeta_1\omega_1\dot{x}_i - \omega_1^2 x_i - \ddot{r}_{i,x} - \left(\frac{\omega_1^2}{g}x_i\right)\ddot{r}_{i,z}\\ -2\zeta_1\omega_1\dot{y}_i - \omega_1^2 y_i - \ddot{r}_{i,y} - \left(\frac{\omega_1^2}{g}y_i\right)\ddot{r}_{i,z}\end{bmatrix} \qquad i = IC, EC. \tag{4.4b}$$

The constraints in (4.3c, 4.3d) define the initial and final states of **x**. In (4.3d), it can be observed that the symbols "-" appear in correspondence of $\dot{x}_{IC}(t_{end})$, $\dot{y}_{IC}(t_{end})$ and $\dot{x}_{EC}(t_{end})$, $\dot{y}_{EC}(t_{end})$, indicating that no final condition was imposed on these variables. The constraint ensuring that the sloshing height of the external containers remains below $\overline{\eta}_{lim}$ is represented by the inequality constraint in (4.3e), while (4.3f) limits the residual liquid oscillations to $0.2\overline{\eta}_{lim}$ after the trajectory time t_{end}. Condition (4.3g) constrains the maximum absolute value that each component of **u** can assume. The waypoints are imposed in equation (4.3h), where n_p represents the number of selected points[1], the latter being represented by $\mathbf{r}_j$. No condition is imposed on θ at time t_j. Condition (4.3i) ensures that the computed path does not assume forbidden vertical positions.

4.2.2 Considered trajectories and selected points

Referring to the paths shown in figures 3.10 and 3.11, figures 4.2 and 4.3 display the same paths, on which specific points are selected, representing the waypoints for the point-to-point optimizations, i.e., the $\mathbf{r}_j$ of equation (4.3h). These waypoints are selected along the original paths to ensure that the optimized paths remain similar to the initial ones, thereby enabling a meaningful comparison of the results.
The considered trajectories are defined as follows:

- *Trajectory Points 1A:* determined by the optimizer based on the waypoint on Path 1 (highlighted in the figure 4.2) and by imposing $\theta_0 = 0$, $\theta_{end} = \pi$;
- *Trajectory Points 1B:* determined by the optimizer based on the waypoint on Path 1 (highlighted in the figure 4.2) and by imposing $\theta_0 = 0$, $\theta_{end} = 2\pi$;
- *Trajectory Points 2:* determined by the optimizer based on the 8 waypoints on Path 2 (highlighted in the figure 4.3) and by imposing $\theta_0 = 0$, $\theta_{end} = \pi$.

[1]The waypoints in (4.3h) do not include the initial and final points of the trajectory, which are imposed by (4.3c) and (4.3d).

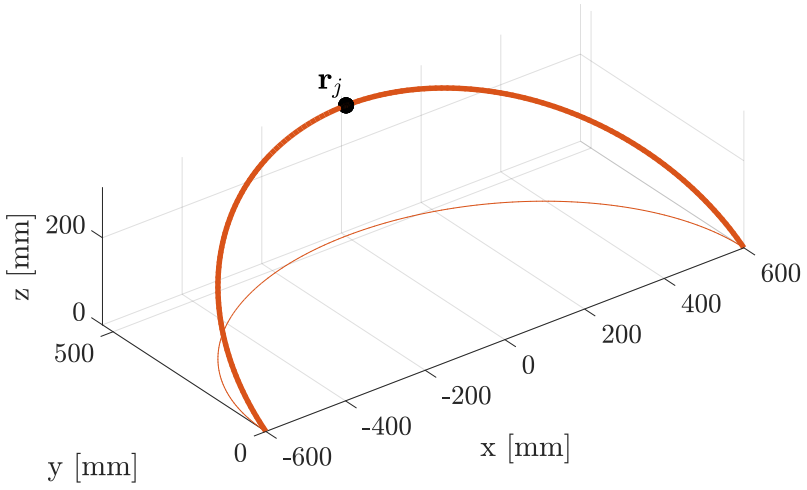


Figure 4.2: Waypoints on Path 1.

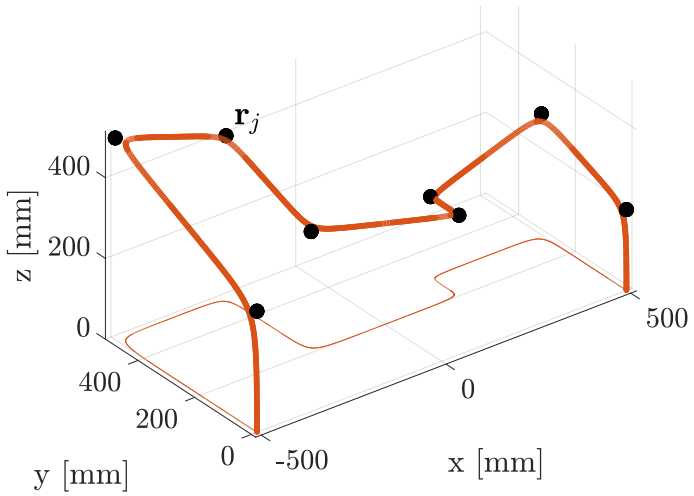


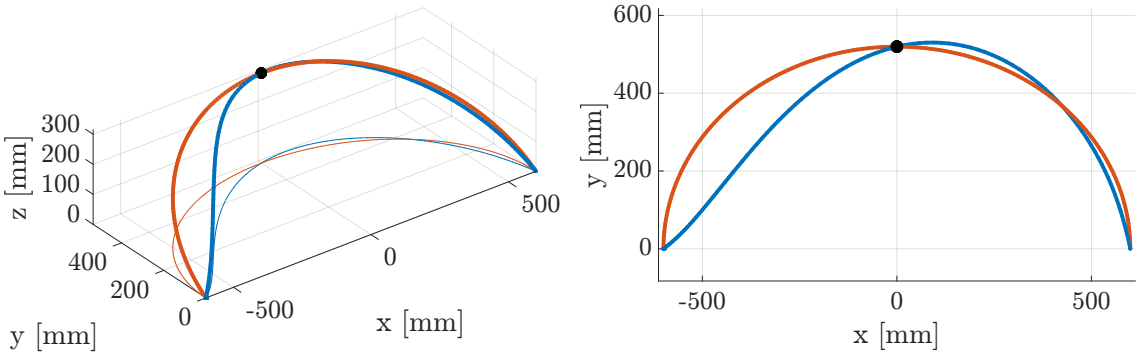
Figure 4.3: Waypoints on Path 2.

4.2.3 Optimization results

This section presents the results from the optimizations obtained by considering the three container configurations (figures 3.9), the three aforementioned trajectories, and by constraining only the sloshing height of IC and EC.
In addition to the graphs representing the sloshing height trends, which are reported as in chapter 3, the figures related to the optimized paths are also provided. Indeed, as repeatedly emphasized, unlike the optimizations proposed in chapter 3, the optimizations for point-to-point motion provide, in addition to the optimal motion law, the optimal 3D path and the optimal trend of θ. Therefore, the optimal paths obtained from the assigned waypoints are presented below[2] together with the assigned paths, allowing for a comparison of their differences, specifically: figure 4.4 shows the comparison between the assigned path of Trajectory 1A (section 3.4) in red and the optimal path of Trajectory Points 1A (section 4.2.2) in blue; figure 4.5 shows the comparison between the assigned path of Trajectory 1B (section 3.4) in red and the optimal path of Trajectory Points 1B (section 4.2.2) in blue; figure 4.6 shows the comparison between the assigned path of Trajectory 2 (section 3.4) in red and the optimal path of Trajectory

[2]To avoid redundancy, only the graphs related to the configuration with 4 containers are shown; very similar results are obtained with 6 and 8 containers as well.

Points 2 (section 4.2.2) in blue. Observing these figures, one can notice the strategies applied by the optimizer in order to reduce the motion duration and limit the sloshing heights. In particular, from figures 4.4 and 4.5, it is evident that the optimizer seeks to connect the central waypoint with the initial and final positions via the shortest possible path. This results in an optimized path that is shorter than the initial one, which can be traversed more quickly. Regarding figure 4.6, it is observed that, for this path, the optimizer adopts an opposite strategy compared to the previous cases, namely, the optimized path is longer than the initial one. This is due to the fact that the initial path, although shorter, presents abrupt changes in direction, i.e., very tight curves. Traversing these tight curves at high speeds induces significant centripetal accelerations on the containers, which consequently result in increased sloshing. To mitigate this effect, the optimizer determines a path that, at the cost of an increased length, presents smoother and more rounded curves, which can be traversed more rapidly without exciting the containers with high centrifugal forces. For completeness, the optimized trends of θ, $\dot{\theta}$, and $\ddot{\theta}$ for these three cases are also reported. Specifically, figure 4.7a shows the trends related to the path in figure 4.4; figure 4.7b shows the trends related to the path in figure 4.5; finally, figure 4.7c shows the trends related to the path in figure 4.6. It can be observed that, as expected, the higher the ratio between $\Delta\theta = (\theta_{end} - \theta_0)$ and the path length, the greater the maximum angular accelerations $\ddot{\theta}$. It is interesting to observe the strategy adopted by the optimizer in defining the trends in figure 4.7c. It can be seen that the rotation is almost entirely performed in the first part of the motion, remaining nearly constant in the second part, which results in a purely translational motion of the tray.



(a) 3D view.

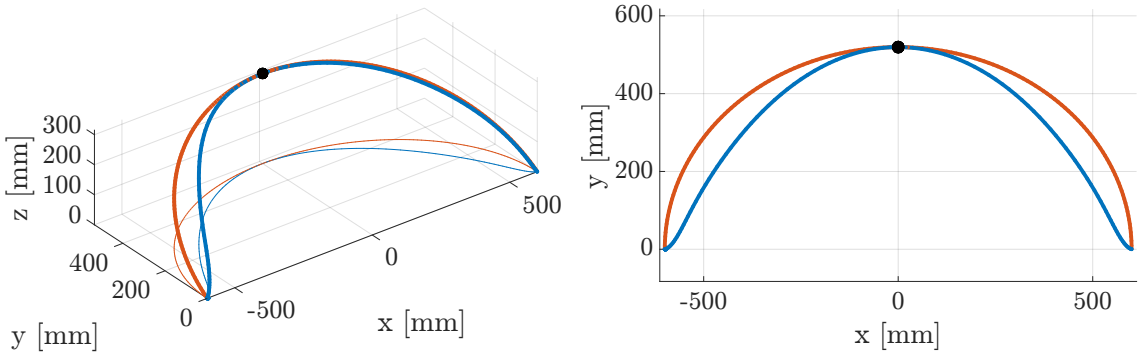
(b) Top view.

Figure 4.4: Comparison of Trajectory Points 1A, 4 containers case.

Regarding the graphs of the sloshing-height trends:

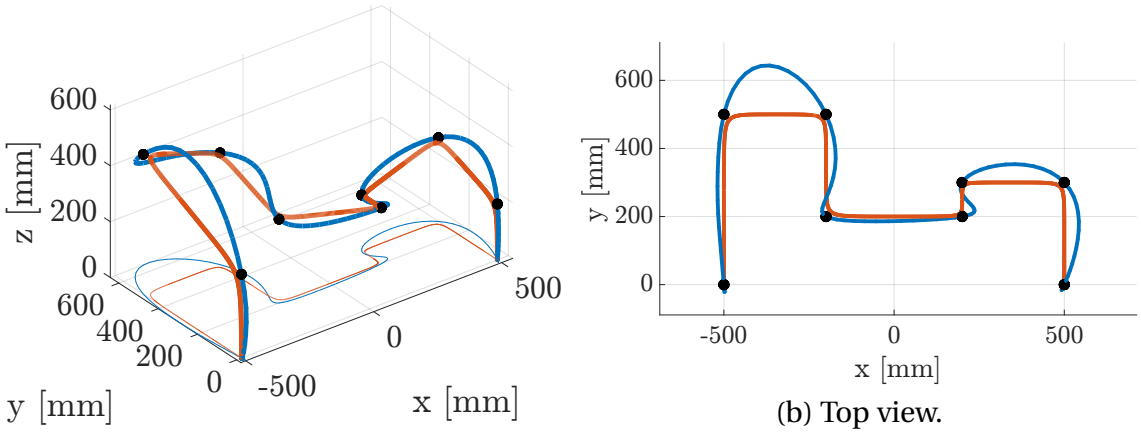
- figures 4.8 show the sloshing-height trends for Trajectory Points 1A;
- figures 4.9 show the sloshing-height trends for Trajectory Points 1B;
- figures 4.10 show the sloshing-height trends for Trajectory Points B.

It can be observed that, even in this case, constraining the sloshing heights only for IC and EC proves to be a successful choice, as the sloshing-height limits for each container are satisfied. The aforementioned figures also report the duration of the optimized motion t_{end} and the computation time t_{calc}. For convenience, these times are



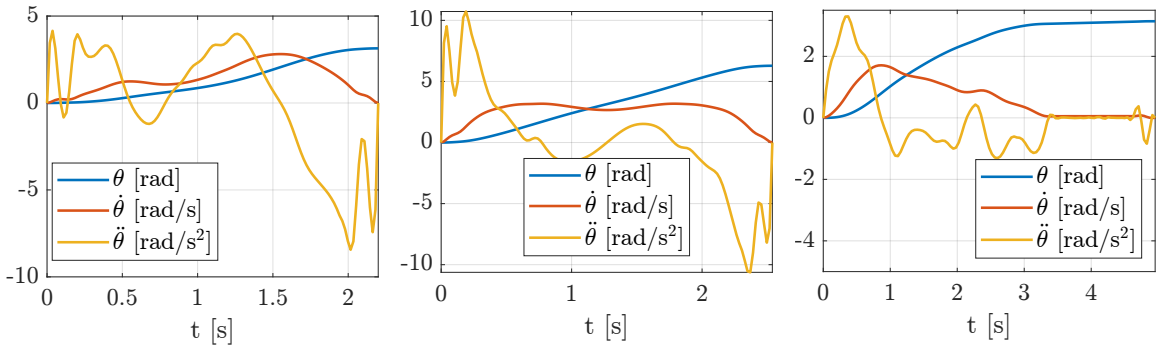
(a) 3D view. (b) Top view.

Figure 4.5: Comparison of Trajectory Points 1B, 4 containers case.



(a) 3D view. (b) Top view.

Figure 4.6: Comparison of Trajectory Points B, 4 containers case.



(a) Trajectory Points 1A. (b) Trajectory Points 1B. (c) Trajectory Points 2.

Figure 4.7: Comparison of θ trends.

summarized in the following tables, which also include the times related to the optimizations with assigned paths from chapter 3, allowing for a comparison and the determination of the optimal strategy. Specifically:

- table 4.1 presents the times for *Trajectory 1A*;
- table 4.2 presents the times for *Trajectory 1B*;

- table 4.3 presents the times for *Trajectory 2*;

Table 4.1: Comparison Trajectories 1A.

Optimization type	**4 containers**	**6 containers**	**8 containers**
Assigned path	$t_{end} = 2.28s$	$t_{end} = 2.45s$	$t_{end} = 2.63s$
	$t_{calc} = 11.89s$	$t_{calc} = 12.45s$	$t_{calc} = 13.02s$
Assigned points	$t_{end} = 2.20s$	$t_{end} = 2.37s$	$t_{end} = 2.53s$
	$t_{calc} = 16.22s$	$t_{calc} = 18.19s$	$t_{calc} = 22.11s$

Table 4.2: Comparison Trajectories 1B.

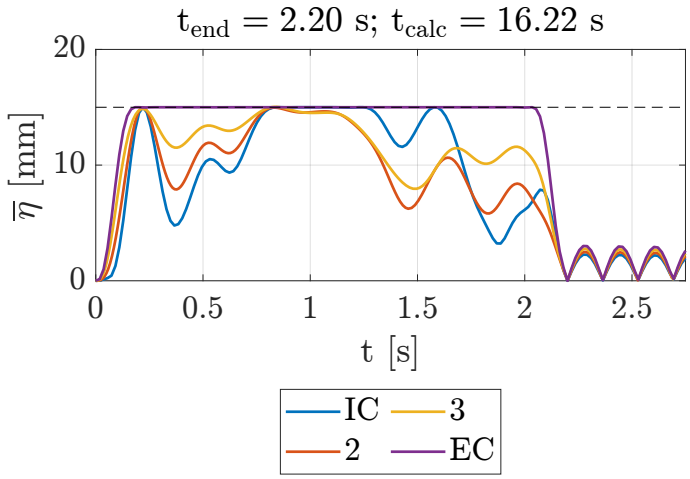
Optimization type	**4 containers**	**6 containers**	**8 containers**
Assigned path	$t_{end} = 2.87s$	$t_{end} = 3.33s$	$t_{end} = 3.71s$
	$t_{calc} = 11.75s$	$t_{calc} = 10.18s$	$t_{calc} = 13.18s$
Assigned points	$t_{end} = 2.54s$	$t_{end} = 2.95s$	$t_{end} = 3.33s$
	$t_{calc} = 20.23s$	$t_{calc} = 25.23s$	$t_{calc} = 69.56s$

Table 4.3: Comparison Trajectories 2.

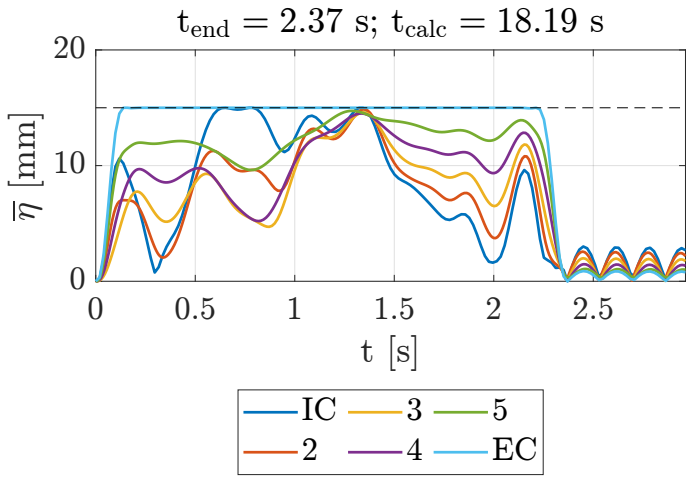
Optimization type	**4 containers**	**6 containers**	**8 containers**
Assigned path	$t_{end} = 4.73s$	$t_{end} = 4.87s$	$t_{end} = 5.01s$
	$t_{calc} = 89.70s$	$t_{calc} = 71.57s$	$t_{calc} = 129.89s$
Assigned points	$t_{end} = 4.95s$	$t_{end} = 4.95s$	$t_{end} = 4.95s$
	$t_{calc} = 20.19s$	$t_{calc} = 18.48s$	$t_{calc} = 15.33s$

From tables 4.1 and 4.2, it can be observed that the point-to-point optimization leads to a slight but beneficial reduction of t_{end}. However, it is important to note that, unlike the first two cases, in table 4.3, optimizing the path with assigned waypoints does not offer any advantage compared to optimizing with a predefined path. In fact, it even results in a higher t_{end} for the cases with 4 and 6 containers. This can be attributed to the fact that, following the formulation of this problem in section 4.2.1, the times t_j at which the waypoints must be reached are predetermined, and these assigned times may not be the optimal times for reaching the positions $\mathbf{r}_j$. Therefore, by imposing both the waypoint and the time for the passage, we are limiting the optimizer. This limitation becomes more pronounced as the number of imposed waypoints increases,

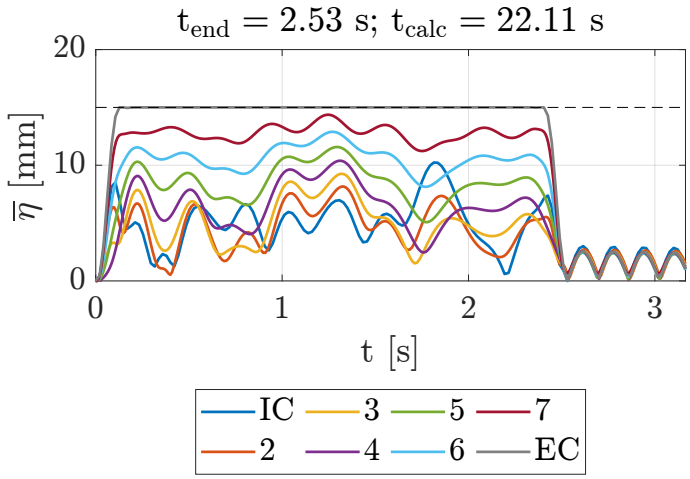
which is why for trajectories 1A and 1B, where only one waypoint is imposed, this effect is not evident. To address this limitation, the idea is to impose way-volumes instead of waypoints, hence giving the optimizer more freedom in determining the exact position in space at time t_j. This new strategy is presented in section 4.3.



(a) 4 containers.

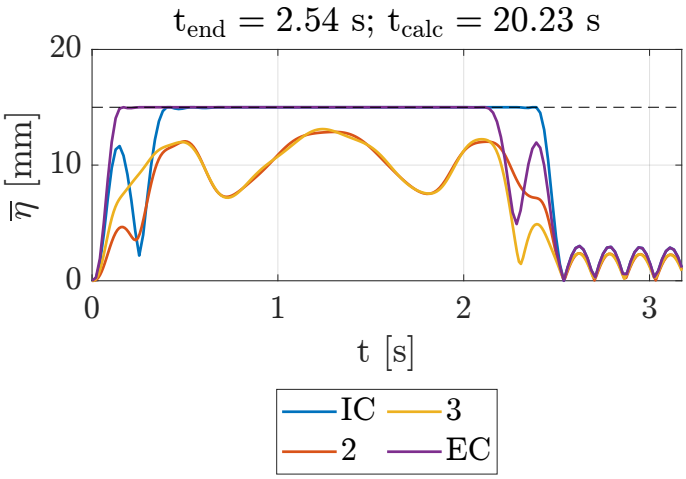


(b) 6 containers.

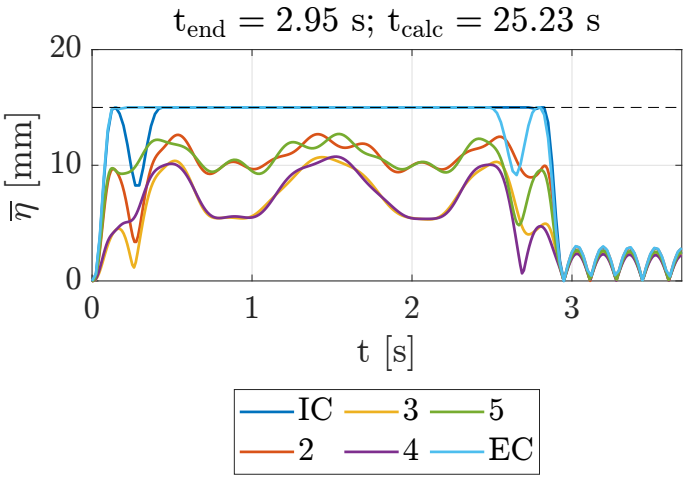


(c) 8 containers.

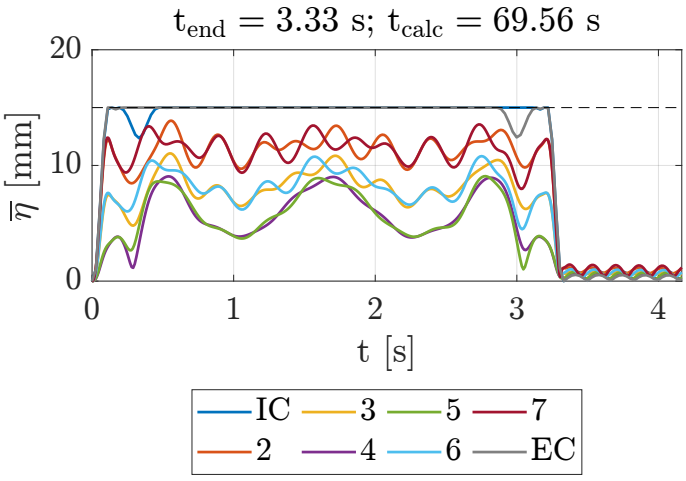
Figure 4.8: Sloshing heights for Trajectory Points 1A.



(a) 4 containers.

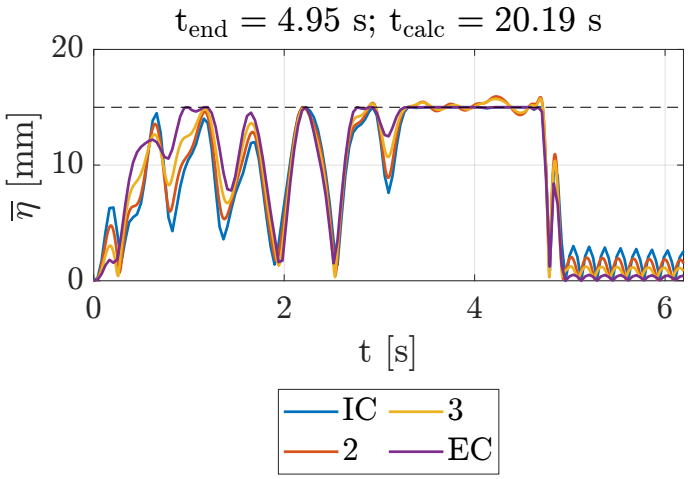


(b) 6 containers.

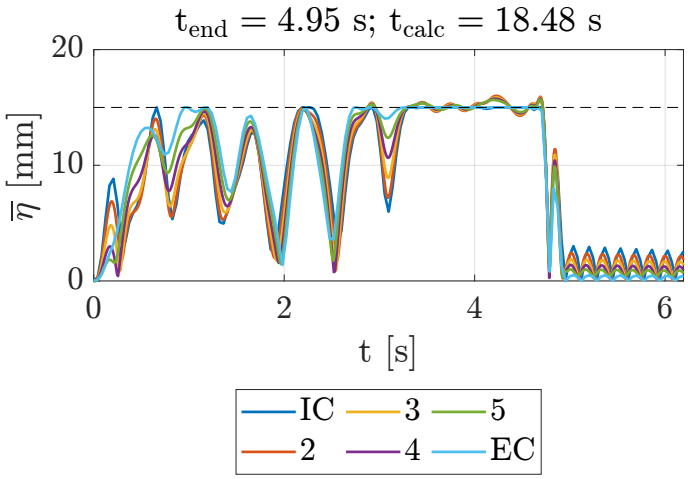


(c) 8 containers.

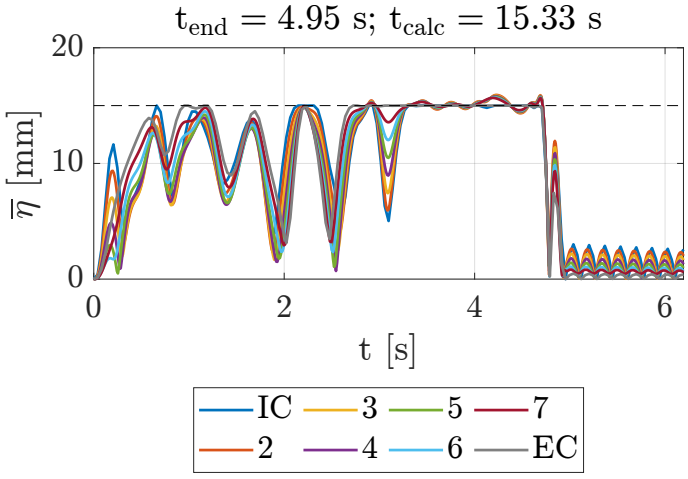
Figure 4.9: Sloshing heights for Trajectory Points 1B.



(a) 4 containers.



(b) 6 containers.



(c) 8 containers.

Figure 4.10: Sloshing heights for Trajectory Points 2.

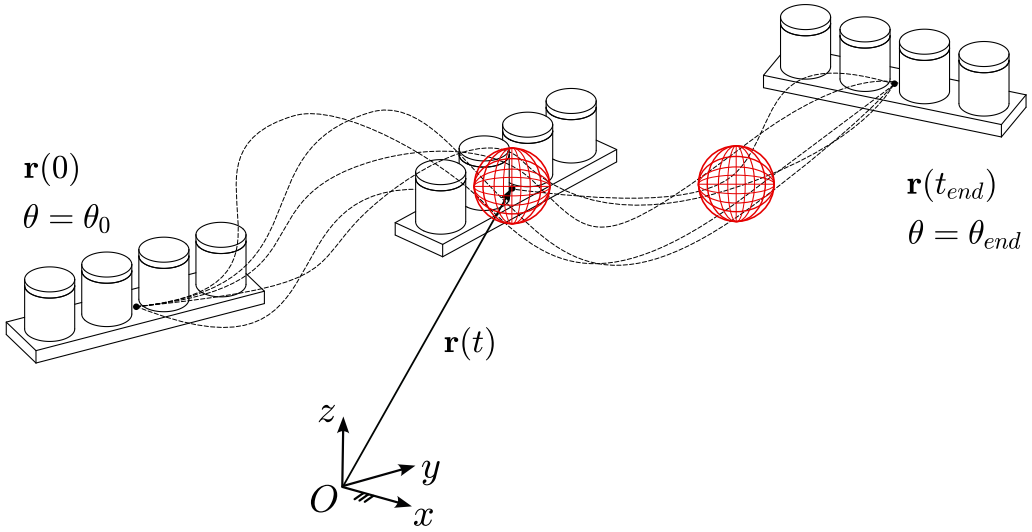


Figure 4.11: Point-to-point with assigned way-volumes generic trajectories.

4.3 Assigned way-volumes

Figure 4.11 shows a graphical representation of the problem at hand. It can be observed that multiple paths are possible, as in these optimizations only meaningful way-volumes (highlighted in red in the figure) are assigned and the path is free.

4.3.1 Problem formulation

The problem formulation is identical to the one presented in section 4.2.1, except for condition (4.3h), which is modified as follows:

$$\left(\mathbf{r}(t_j) - \mathbf{r}_j\right)^T \left(\mathbf{r}(t_j) - \mathbf{r}_j\right) \leq \rho^2 \qquad j = 1, ..., n_v; \quad t_j \in (t_0, t_{end}), \tag{4.5}$$

where n_v represents the number of way-volumes. Spheres have been utilized, with ρ denoting the radius set to $\rho = 50$ mm.

4.3.2 Considered trajectories and selected volumes

Referring to the paths in figures 3.10 and 3.11, figures 4.12 and 4.13 illustrate the same paths now provided with specific volumes, which represent the way-volumes for the optimizations. As in section 4.2.2, these way-volumes are selected along the original paths to ensure that the optimized paths remain similar to the initial ones, thereby enabling a meaningful comparison of the results.

The considered trajectories are defined as follows:

- *Trajectory Volumes 1A:* determined by the optimizer based on the way-volume on Path 1 and by imposing $\theta_0 = 0$, $\theta_{end} = \pi$;
- *Trajectory Volumes 1B:* determined by the optimizer based on the way-volume on Path 1 and by imposing $\theta_0 = 0$, $\theta_{end} = 2\pi$;
- *Trajectory Volumes 2:* determined by the optimizer based on the 8 way-volumes on Path 2 and by imposing $\theta_0 = 0$, $\theta_{end} = \pi$.

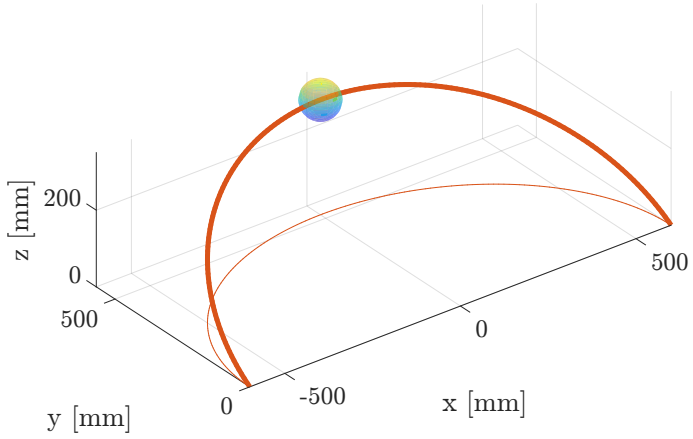


Figure 4.12: Way-volumes on Path 1.

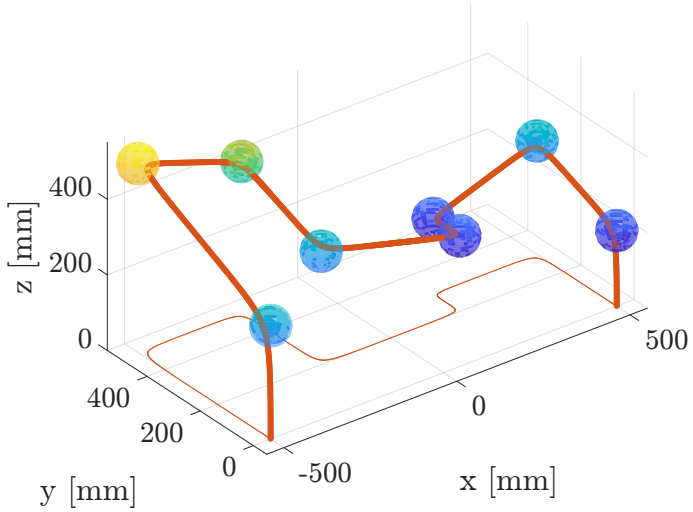


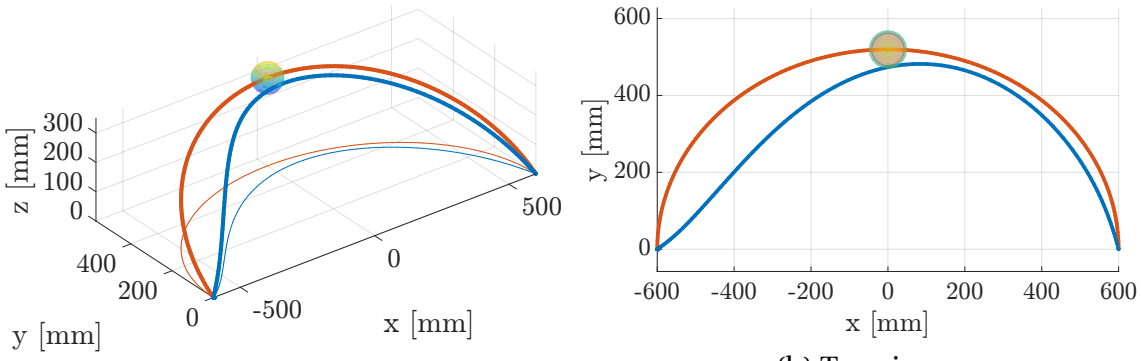
Figure 4.13: Way-volumes on Path 2.

4.3.3 Optimization results

As in section 4.2.3, this section presents the results from the optimizations obtained by considering the three container configurations (figures 3.9), the three trajectories just described, and by constraining only the sloshing height of IC and EC.
The optimal paths obtained from the assigned way-volumes are shown[3], specifically: figure 4.14 shows the comparison between the assigned path of Trajectory 1A (section 3.4) in red and the optimal path of Trajectory Points 1A (section 4.3.2) in blue; figure 4.15 shows the comparison between the assigned path of Trajectory 1B (section 3.4) in red and the optimal path of Trajectory Points 1B (section 4.3.2) in blue; figure 4.16 shows the comparison between the assigned path of Trajectory 2 (section 3.4) in red and the optimal path of Trajectory Points 2 (section 4.3.2) in blue. As in the previous section, the strategies adopted by the optimizer to reduce t_{end} in the definition of the optimized trajectory can be observed. In particular, for the cases shown in figures 4.14 and 4.15, it can be seen that the optimal trajectory is tangent at the innermost point of the imposed way-volume. This ensures that the optimized path is as short as possible, allowing it to be traversed in a shorter time. In the case shown in figure 4.16, exactly as

[3]To avoid redundancy, only the graphs related to the configuration with 4 containers are shown, very similar results are obtained with 6 and 8 containers as well.

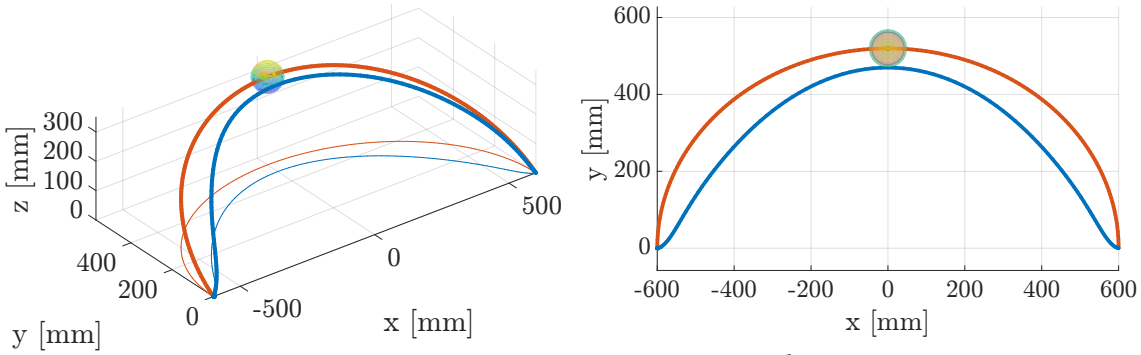
in the previous section, the optimizer adopts another strategy: it defines an optimized path that is longer than the initial one. Despite its increased length, this new path presents smoother curves, which result in lower centripetal accelerations compared to the tight turns of the initial path. Consequently, these smoother curves allow for higher traversal velocity. For completeness, the optimized trends of θ, $\dot{\theta}$, and $\ddot{\theta}$ for these three cases are also reported. Specifically, figure 4.17a shows the trends related to the path in figure 4.14; figure 4.17b shows the trends related to the path in figure 4.15; finally, figure 4.17c shows the trends related to the path in figure 4.16. It can be observed that, as expected, the higher the ratio between $\Delta\theta = (\theta_{end} - \theta_0)$ and the path length, the greater the maximum angular accelerations $\ddot{\theta}$. It is interesting to observe the strategy adopted by the optimizer in defining the trends in figure 4.17c. As in section 4.2.3 it can be seen that the rotation is almost entirely performed in the first part of the motion, remaining nearly constant in the second part, which results in a purely translational motion of the tray.



(a) 3D view.

(b) Top view.

Figure 4.14: Comparison of Trajectory Volumes 1A, 4 containers case.



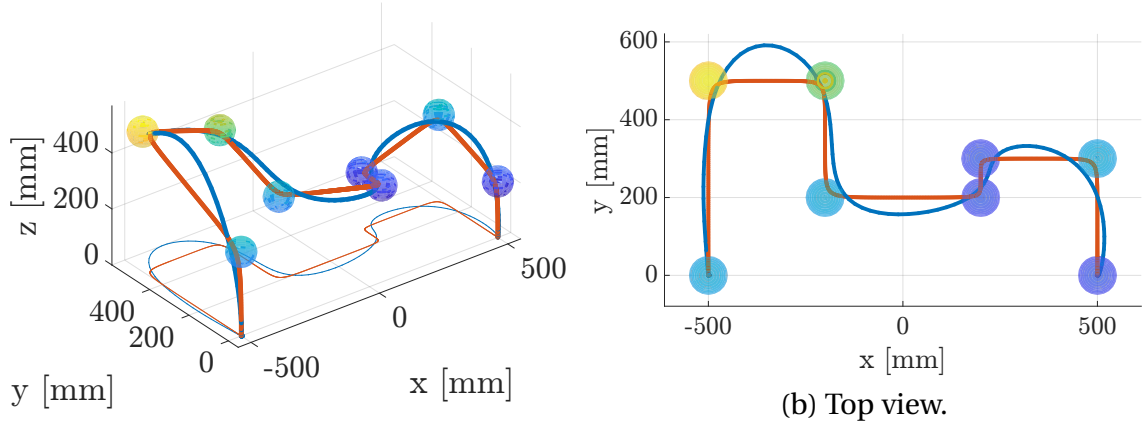
(a) 3D view.

(b) Top view.

Figure 4.15: Comparison of Trajectory Volumes 1B, 4 containers case.

The graphs of the sloshing-height trends are shown, in particular:

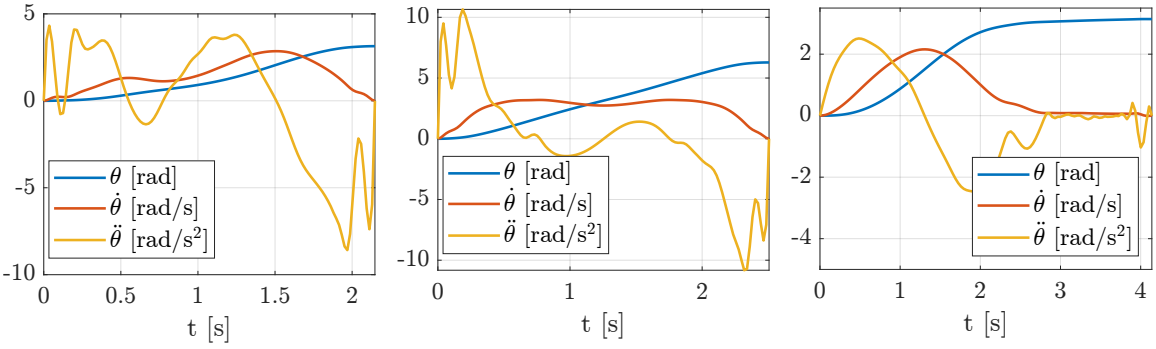
- figure 4.18 shows the sloshing-height trends for Trajectory Volumes 1A;
- figure 4.19 shows the sloshing-height trends for Trajectory Volumes 1B;
- figure 4.20 shows the sloshing-height trends for Trajectory Volumes 2.



(a) 3D view.

(b) Top view.

Figure 4.16: Comparison of Trajectory Volumes B, 4 containers case.



(a) Trajectory Volumes 1A.

(b) Trajectory Volumes 1B.

(c) Trajectory Volumes 2.

Figure 4.17: Comparison of θ trends.

Observing the trends of the sloshing heights, it can be seen that, even for this type of optimization, optimizing only the external containers allows compliance with the maximum sloshing-height limit for each container. It is important to analyze the t_{end} of the optimizations, which are reported along with the computation times in the sloshing-height graphs. Indeed, in the previous section, it was shown that the point-to-point optimization with assigned waypoints was not advantageous for *Trajectory 2*, as assigning both the waypoints and the passage times t_j limited the optimizer. This limitation has been addressed in this section by assigning volumes instead of points. Let us now analyze the obtained times. For comparison purposes, the times related to the optimizations with assigned paths and point-to-point optimization with assigned waypoints are also reported in the following tables, specifically:

- table 4.4 presents the times for *Trajectories 1A*;
- table 4.5 presents the times for *Trajectories 1B*;
- table 4.6 presents the times for *Trajectories 2*.

From tables 4.4 and 4.5, it can be observed that optimizing the path by assigning way-volumes always leads to a reduction in trajectory duration (t_{end}) compared to the case with an assigned path, the same applies comparing the way-volume case with the case

Table 4.4: Comparison Trajectories 1A.

Optimization type	**4 containers**	**6 containers**	**8 containers**
Assigned path	$t_{end} = 2.28s$	$t_{end} = 2.45s$	$t_{end} = 2.63s$
	$t_{calc} = 11.89s$	$t_{calc} = 12.45s$	$t_{calc} = 13.02s$
Assigned points	$t_{end} = 2.20s$	$t_{end} = 2.37s$	$t_{end} = 2.53s$
	$t_{calc} = 16.22s$	$t_{calc} = 18.19s$	$t_{calc} = 22.11s$
Assigned volumes	$t_{end} = 2.15s$	$t_{end} = 2.33s$	$t_{end} = 2.49s$
	$t_{calc} = 16.68s$	$t_{calc} = 15.11s$	$t_{calc} = 13.63s$

Table 4.5: Comparison Trajectories 1B.

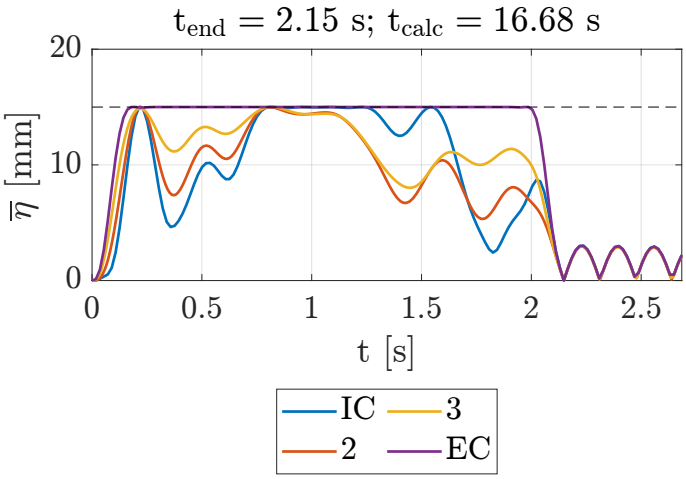
Optimization type	**4 containers**	**6 containers**	**8 containers**
Assigned path	$t_{end} = 2.87s$	$t_{end} = 3.33s$	$t_{end} = 3.71s$
	$t_{calc} = 11.75s$	$t_{calc} = 10.18s$	$t_{calc} = 13.18s$
Assigned points	$t_{end} = 2.54s$	$t_{end} = 2.95s$	$t_{end} = 3.33s$
	$t_{calc} = 20.23s$	$t_{calc} = 25.23s$	$t_{calc} = 69.56s$
Assigned volumes	$t_{end} = 2.51s$	$t_{end} = 2.93s$	$t_{end} = 3.32s$
	$t_{calc} = 22.00s$	$t_{calc} = 49.70s$	$t_{calc} = 44.94s$

Table 4.6: Comparison Trajectories 2.

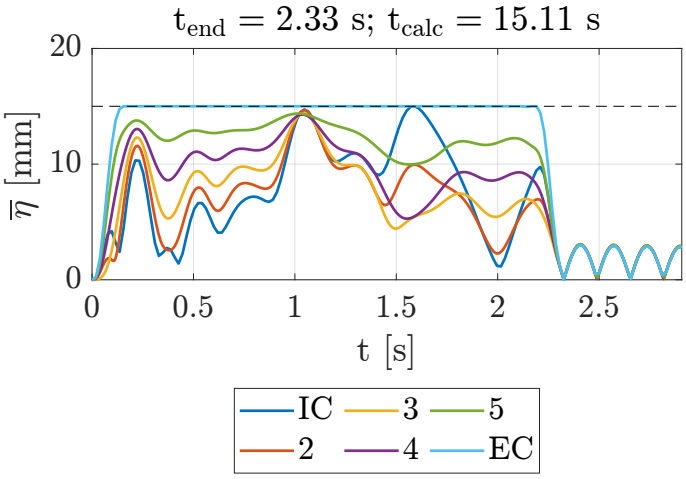
Optimization type	**4 containers**	**6 containers**	**8 containers**
Assigned path	$t_{end} = 4.73s$	$t_{end} = 4.87s$	$t_{end} = 5.01s$
	$t_{calc} = 89.70s$	$t_{calc} = 71.57s$	$t_{calc} = 129.89s$
Assigned points	$t_{end} = 4.95s$	$t_{end} = 4.95s$	$t_{end} = 4.95s$
	$t_{calc} = 20.19s$	$t_{calc} = 18.48s$	$t_{calc} = 15.33s$
Assigned volumes	$t_{end} = 4.14s$	$t_{end} = 4.14s$	$t_{end} = 4.14s$
	$t_{calc} = 26.55s$	$t_{calc} = 15.73s$	$t_{calc} = 18.03s$

of point-to-point optimization with assigned waypoints, although the time reduction is negligible for Trajectory 1A and 1B. This difference is even more pronounced in table 4.6, where, in the case with 8 containers, optimizing the path with assigned way-

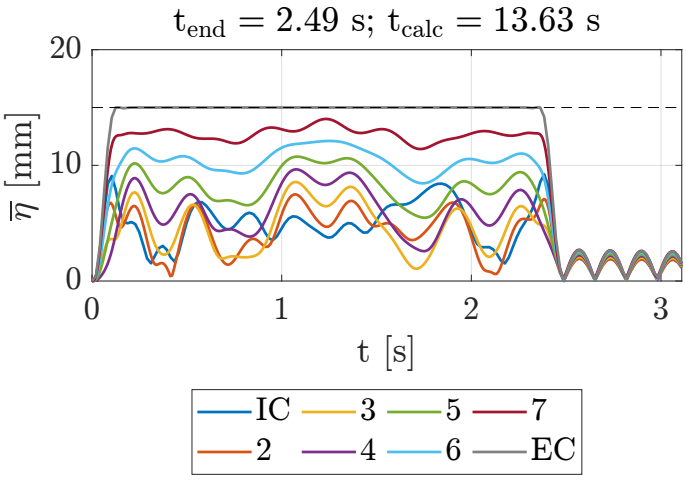
volumes results in a 17.37% reduction in t_{end} compared to the first optimization, and in a 16.36% reduction in t_{end} compared to the point-to-point with assigned waypoint optimization. It can therefore be concluded that point-to-point optimization with assigned way-volumes not only provides a better solution in cases where optimization with assigned waypoints failed (e.g., in the case of Trajectory 2), but it achieves this result with a significant margin. Therefore, point-to-point optimization with assigned way-volumes appears to be the most advantageous strategy to adopt when no predefined path is assigned.



(a) 4 containers.

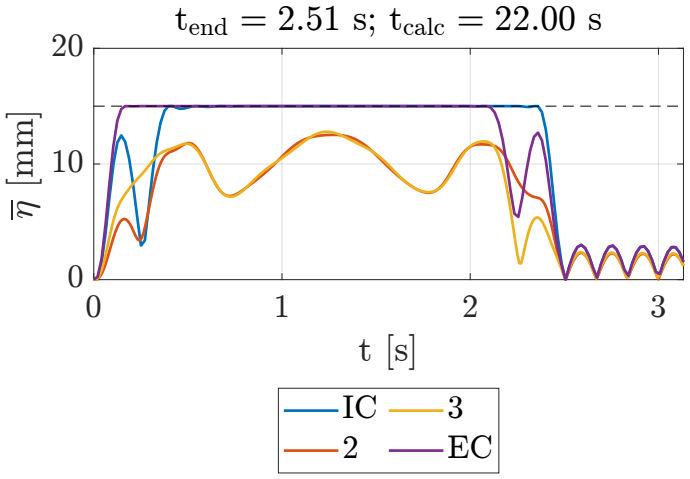


(b) 6 containers.

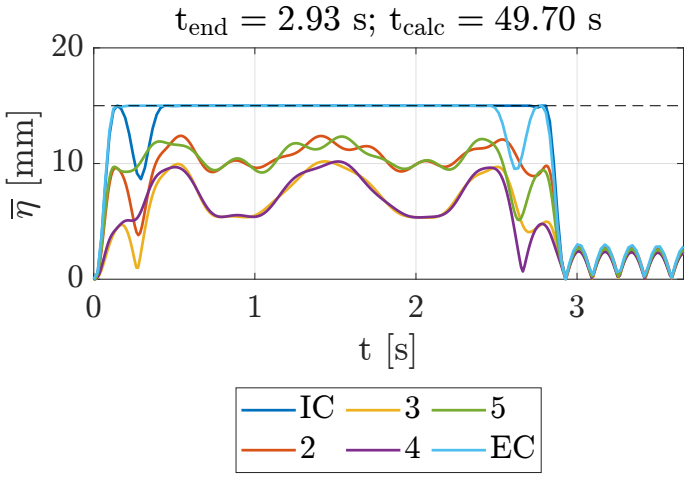


(c) 8 containers.

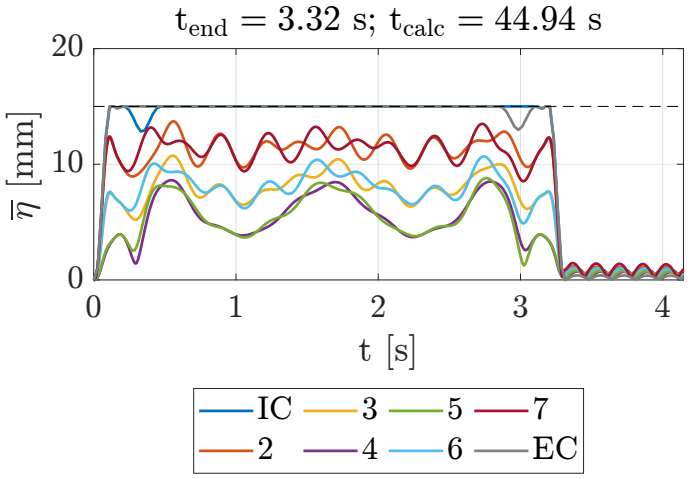
Figure 4.18: Sloshing heights for Trajectory Volumes 1A.



(a) 4 containers.

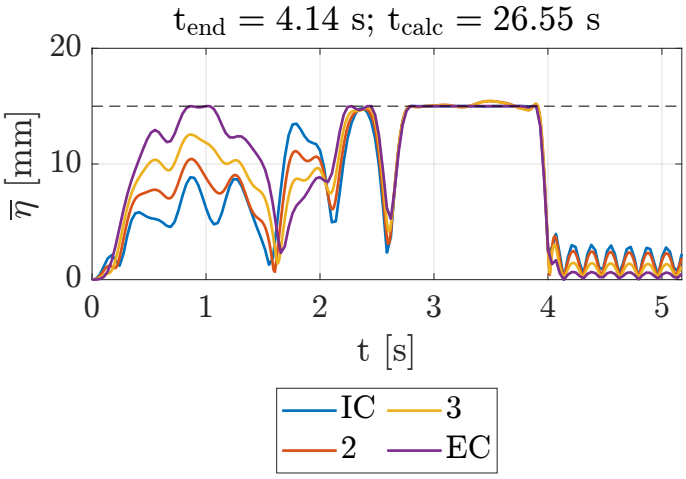


(b) 6 containers.

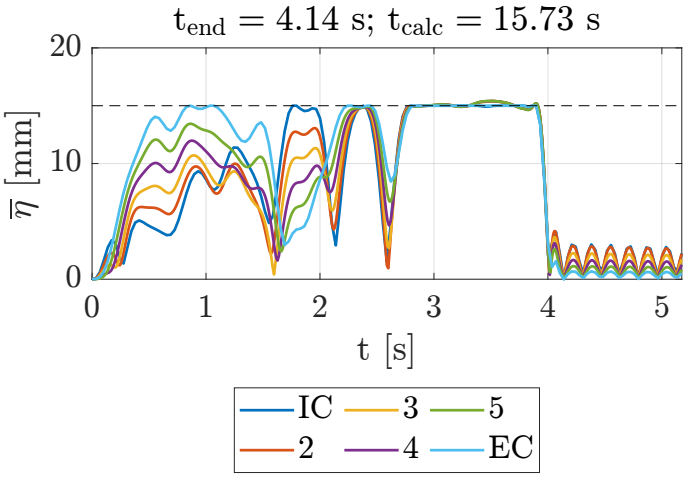


(c) 8 containers.

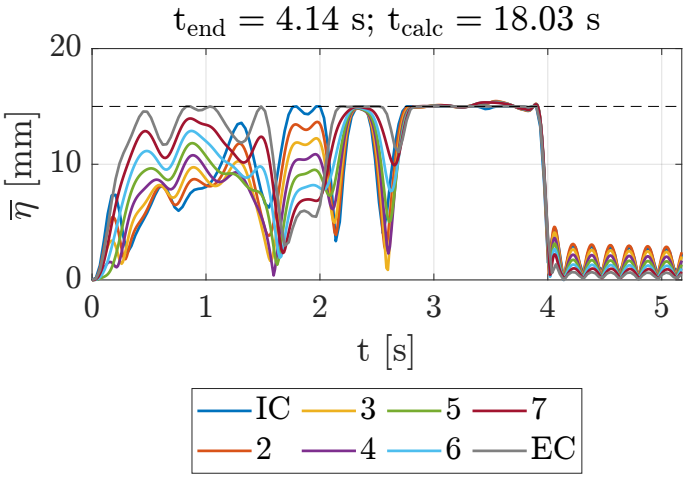
Figure 4.19: Sloshing heights for Trajectory Volumes 1B.



(a) 4 containers.



(b) 6 containers.



(c) 8 containers.

Figure 4.20: Sloshing heights for Trajectory Volumes 2.

Chapter 5

Experimental campaign

This chapter presents the experimental analysis conducted to validate the optimizations described in the previous chapters. The geometry of the selected components and the robot used to execute the trajectories will be introduced. The optimization algorithms employed to generate the robot trajectories will be provided, and finally, the experimental results will be presented.

5.1 Experimental setup

For the experimental tests, containers with a radius $R = 35$ mm were selected, and a static liquid height of $h = 40$ mm was chosen. A modular support structure was designed, consisting of commercial components and parts produced via rapid prototyping, capable of holding $n_c = 8$ containers, as shown in figure 5.1. The support was designed so that the center-to-center distance between two adjacent containers is $i_c = 90$ mm. The support presents an interface capable of engaging with the robot *end-effector* (EE). The robot chosen for the trajectory execution is a 6-DOF serial manipulator, specifically a *Comau SMART-SiX* (see figure 5.2). To determine the sloshing height of the liquid during motion, the videos recorded by two *GoPro Hero8* cameras were used, positioned as shown in figure 5.3. As seen in the figure, to correctly position one of the cameras and ensure a correct recording, the two containers adjacent to the one being recorded had to be removed. By appropriately repositioning the other GoPro and removing two different containers, it is possible to correctly record all the containers. Additionally, two white backgrounds were mounted on the opposite sides, relative to the GoPro cameras, of the container; these elements were necessary to isolate the liquid in the video frames and prevent background objects from interfering with the detection of the liquid.

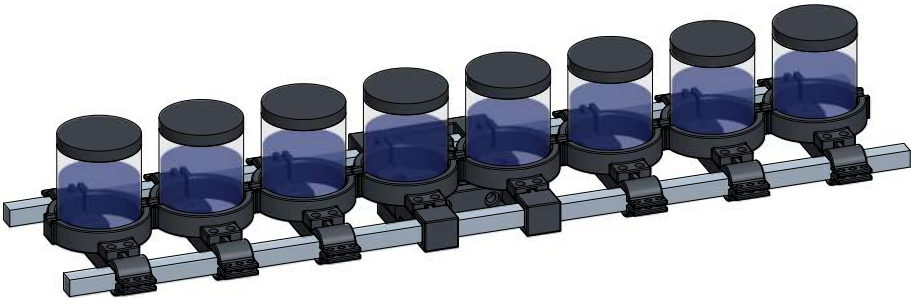


Figure 5.1: Multi-container setup.

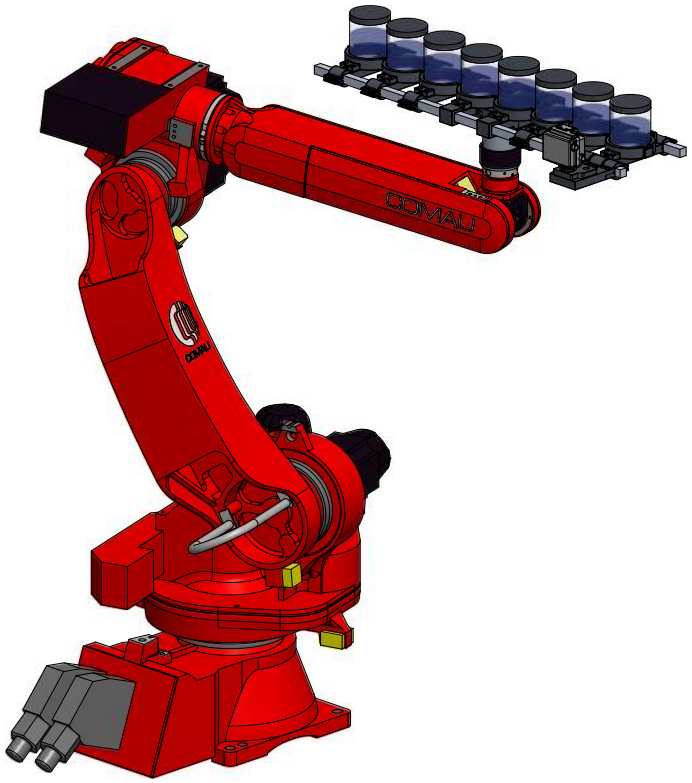


Figure 5.2: Comau SMART-SiX.

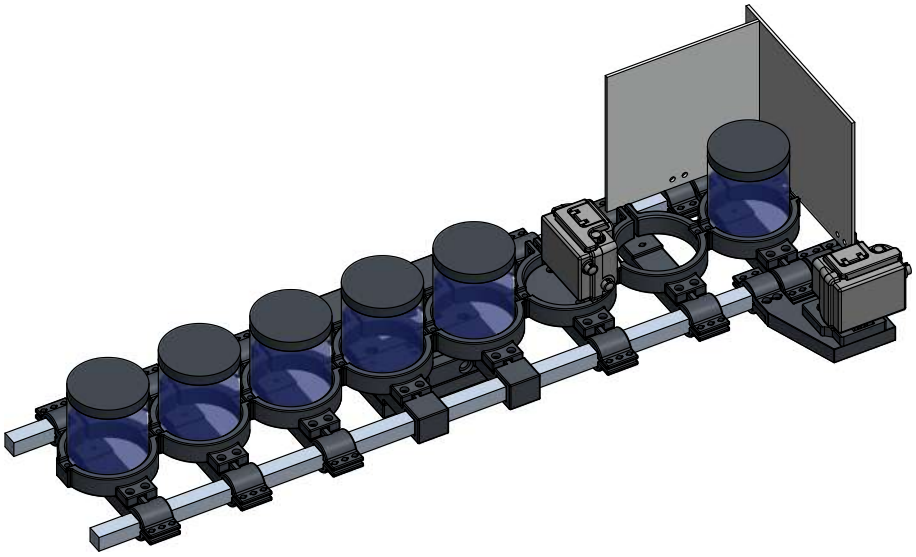


Figure 5.3: GoPro setup.

5.2 Problem formulations for the experiments

This section illustrates the algorithms implemented for experimental purposes. Based on the results reported in chapter 3, simulations are carried out by constraining only the sloshing height of the outer containers IC and EC. Differently from the problem formulations discussed in the previous chapters, in those presented in this section, the joint positions and velocities of the robot will be included among the problem constraints. The robot, in fact, through the appropriate movement of its six joints, allows control over the position and orientation of the support. However, there are certain limitations:

- position limits: the entire desired path must be contained within the robot workspace;
- velocity limits: each joint has its own maximum angular velocity, therefore, the trajectory velocity profile, in order to be executed, must not require joint velocities that are beyond the permissible limits.

In order to appropriately justify the influence of these two aspects on the optimization codes, a brief overview of serial manipulators kinematics is required [45], and it is provided in appendix B.

5.2.1 Assigned-path optimization

The formulation is identical to the one described in section 3.2.2, with the difference that, as stated above, only the outer containers IC and EC are considered in the state vector and in the constraints. Considering the control input[1]:

$$u = \dddot{s}, \tag{5.1}$$

the system state is defined by a vector $\mathbf{x} \in \mathbb{R}^{11}$, namely[2]:

$$\mathbf{x} = [s \;\; \dot{s} \;\; \ddot{s} \;\; x_{IC} \;\; y_{IC} \;\; \dot{x}_{IC} \;\; \dot{y}_{IC} \;\; x_{EC} \;\; y_{EC} \;\; \dot{x}_{EC} \;\; \dot{y}_{EC}]^T. \tag{5.2}$$

The optimization problem can be formulated as:

$$\min_{t_{end},\, u} \left[\int_0^{t_{end}} (1 + ku^2)\, dt \right] \tag{5.3a}$$

[1] As seen in chapter 3, with this control input, θ is defined as a linear function of s, namely: $\theta = \theta_0 + s \cdot (\theta_{end} - \theta_0)$.

[2] The robot joint variables were not included in the state vector to simplify the problem by reducing its dimensions. Instead, they were directly expressed as functions of s and $\dot{s}$ and considered among the constraints.

subject to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) \tag{5.3b}$$

$$\mathbf{x}(0) = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \tag{5.3c}$$

$$\mathbf{x}(t_{end}) = [1\ 0\ 0\ 0\ 0\ -\ -\ 0\ 0\ -\ -]^T \tag{5.3d}$$

$$\overline{\eta}_i(t) \leq \overline{\eta}_{lim} \qquad i = IC, EC; \quad t \in [0, t_{end}] \tag{5.3e}$$

$$\overline{\eta}_i(t) \leq 0.2\overline{\eta}_{lim} \qquad i = IC, EC; \quad t > t_{end} \tag{5.3f}$$

$$|u| \leq u_{max} \tag{5.3g}$$

$$|q_j| \leq q_{j,max} \qquad j = 1, ..., 6 \tag{5.3h}$$

$$|\dot{q}_j| \leq \dot{q}_{j,max} \qquad j = 1, ..., 6. \tag{5.3i}$$

The function **f** in (5.3b) includes the integration chain of s from u (5.4a) and the sloshing dynamics of each container (5.4b) and (5.4c), expressed by the NL model equations (2.20), namely:

$$\frac{d}{dt}\begin{bmatrix} s \\ \dot{s} \\ \ddot{s} \end{bmatrix} = \begin{bmatrix} \dot{s} \\ \ddot{s} \\ u \end{bmatrix} \tag{5.4a}$$

$$\frac{d}{dt}\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \qquad i = IC, EC \tag{5.4b}$$

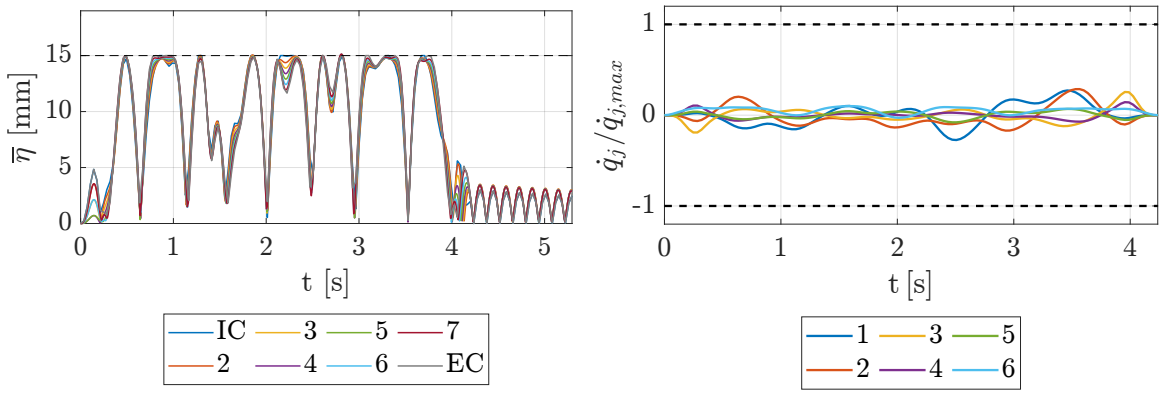
$$\frac{d}{dt}\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} 1 + \frac{C_1^2}{R^2}x_i^2 & \frac{C_1^2}{R^2}x_i y_i \\ \frac{C_1^2}{R^2}x_i y_i & 1 + \frac{C_1^2}{R^2}x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} f_{i,x} \\ f_{i,y} \end{bmatrix} \qquad i = IC, EC \tag{5.4c}$$

where:

$$\begin{aligned} f_{i,x} = &-\frac{C_1^2}{R^2}(\dot{x}_i^2 + \dot{y}_i^2)x_i + (2\dot{\theta}\dot{y}_i + \dot{\theta}^2 x_i + \ddot{\theta}y_i) - \omega_1^2 x_i\left[1 + \alpha_1(x_i^2 + y_i^2)^{w-1}\right] \\ &- 2\omega_1\zeta_1\left[\dot{x}_i + \frac{C_1^2}{R^2}(x_i\dot{x}_i + y_i\dot{y}_i)x_i\right] - \ddot{r}_{i,x}\cos(\theta) - \ddot{r}_{i,y}\sin(\theta) - \ddot{r}_{i,z}\frac{C_1}{R}x_i, \\ f_{i,y} = &-\frac{C_1^2}{R^2}(\dot{x}_i^2 + \dot{y}_i^2)y_i + (-2\dot{\theta}\dot{x}_i + \dot{\theta}^2 y_i - \ddot{\theta}x_i) - \omega_1^2 y_i\left[1 + \alpha_1(x_i^2 + y_i^2)^{w-1}\right] \\ &- 2\omega_1\zeta_1\left[\dot{y}_i + \frac{C_1^2}{R^2}(x_i\dot{x}_i + y_i\dot{y}_i)y_i\right] + \ddot{r}_{i,x}\sin(\theta) - \ddot{r}_{i,y}\cos(\theta) - \ddot{r}_{i,z}\frac{C_1}{R}y_i. \end{aligned} \tag{5.4d}$$

The constraints in (5.3c, 5.3d) define the initial and final states of **x**. In (5.3d), it can be observed that the symbols "-" appear in correspondence of $\dot{x}_{IC}(t_{end})$, $\dot{y}_{IC}(t_{end})$ and $\dot{x}_{EC}(t_{end})$, $\dot{y}_{EC}(t_{end})$, indicating that no final condition has been imposed on these variables. The constraint ensuring that the sloshing height of the external containers remains below $\overline{\eta}_{lim}$ is represented by the inequality constraint in (5.3e), while (5.3f) limits the residual liquid oscillations to $0.2\overline{\eta}_{lim}$ after the trajectory time t_{end}. Condition (5.3g) constrains the maximum absolute value of u. Finally, (5.3h) and (5.3i) represent the two conditions expressed at the beginning of section 5.2. In particular, condition (5.3h) ensures that, throughout the motion, the position of each robot joint remains within permissible limits, thus ensuring that the trajectory stays within the robot workspace. Given s, the optimizer determines the pose of the EE, and by solving the inverse

kinematics, it determines the joint variables $\mathbf{q}$, which must satisfy this constraint. It is important to clarify that ensuring this constraint is not the responsibility of the optimizer. In fact, the trajectory is predetermined, and therefore, the robot joint variables $\mathbf{q}$ are defined *a priori* from the trajectory. Hence, condition (5.3h) is only an indicator that the trajectory has been correctly chosen, and if not satisfied, another suitable trajectory has to be defined. The condition expressed by (5.3i) ensures that the velocities of each joint, determined by the optimizer, assume permissible values. Starting from s and $\dot{s}$, the optimizer determines the linear and angular velocities of the EE, and by using equation (B.10), verifies that the values of $\dot{\mathbf{q}}$ lie within the allowable range.
As an example, figure 5.4 presents the optimization results for Trajectory 2, considering 8 containers and a sloshing height limit of $\overline{\eta}_{lim} = 15$ mm. Specifically, figure 5.4a illustrates the sloshing-height trends for each container, while figure 5.4b displays the normalized joint velocity profiles of the robot. It can be observed that the joint velocities remain significantly below their limits. This occurs because the constraints that reach saturation in the optimization are those on sloshing heights. Indeed, as shown in figure 5.4a, these values frequently reach $\overline{\eta}_{lim}$.



(a) Sloshing-height trends.

(b) Normalized joint velocities.

Figure 5.4: Trajectory 2 results.

5.2.2 Point-to-point motion optimization

Starting from the results of chapter 4, the formulation of the optimization problem for point-to-point motion with assigned passing volumes is presented below. In fact, as shown in section 4.3.3, assigning way-volumes proves to be the best choice for these types of motion. Given the control input:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_r \\ u_\theta \end{bmatrix} = \begin{bmatrix} \dddot{\mathbf{r}} \\ \dddot{\theta} \end{bmatrix} = \begin{bmatrix} \dddot{r}_x \\ \dddot{r}_y \\ \dddot{r}_z \\ \dddot{\theta} \end{bmatrix}, \tag{5.5}$$

the state of the system is represented by the array $\mathbf{x} \in \mathbb{R}^{20}$, namely:

$$\mathbf{x} = [\mathbf{r}^T \ \dot{\mathbf{r}}^T \ \ddot{\mathbf{r}}^T \ \theta \ \dot{\theta} \ \ddot{\theta} \ x_{IC} \ y_{IC} \ \dot{x}_{IC} \ \dot{y}_{IC} \ x_{EC} \ y_{EC} \ \dot{x}_{EC} \ \dot{y}_{EC}]^T. \tag{5.6}$$

The optimization problem can be formulated as:

$$\min_{t_{end},\mathbf{u}} \left[\int_0^{t_{end}} (1 + k_r \mathbf{u}_r^T \mathbf{u}_r + k_\theta u_\theta^2) dt \right] \tag{5.7a}$$

subject to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{5.7b}$$
$$\mathbf{x}(0) = [\ \mathbf{r}_0^T \quad \mathbf{0}^T \ \mathbf{0}^T \quad \theta_0 \quad 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \tag{5.7c}$$
$$\mathbf{x}(t_{end}) = [\mathbf{r}_{end}^T \ \mathbf{0}^T \ \mathbf{0}^T \ \theta_{end} \ 0\ 0\ 0\ 0\ -\ -\ 0\ 0\ -\ -]^T \tag{5.7d}$$
$$\overline{\eta}_i(t) \le \overline{\eta}_{lim} \qquad i = IC, EC; \quad t \in [0, t_{end}] \tag{5.7e}$$
$$\overline{\eta}_i(t) \le 0.2\overline{\eta}_{lim} \qquad i = IC, EC; \quad t > t_{end} \tag{5.7f}$$
$$|\mathbf{u}_r| \le \mathbf{u}_{r,max}; \quad |u_\theta| \le u_{\theta,max} \tag{5.7g}$$
$$\rho^2 \ge \left(\mathbf{r}(t_j) - \mathbf{r}_j\right)^T \left(\mathbf{r}(t_j) - \mathbf{r}_j\right) \qquad j = 1, ..., n_v; \quad t_j \in (t_0, t_{end}) \tag{5.7h}$$
$$r_z \in [r_{z,min}, r_{z,max}] \tag{5.7i}$$
$$|q_j| \le q_{j,max} \qquad j = 1, ..., 6 \tag{5.7j}$$
$$|\dot{q}_j| \le \dot{q}_{j,max} \qquad j = 1, ..., 6. \tag{5.7k}$$

The function $\mathbf{f}$ in (5.7b) includes the integration chain of $\mathbf{r}$ and θ from $\mathbf{u}$ (5.8a) and the sloshing dynamics of each container (5.8b) and (5.8c), expressed by the NL model equations (2.16), namely:

$$\frac{d}{dt}\begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \\ \theta \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \\ \mathbf{u}_r \\ \dot{\theta} \\ \ddot{\theta} \\ u_\theta \end{bmatrix} \tag{5.8a}$$

$$\frac{d}{dt}\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \qquad i = IC, EC \tag{5.8b}$$

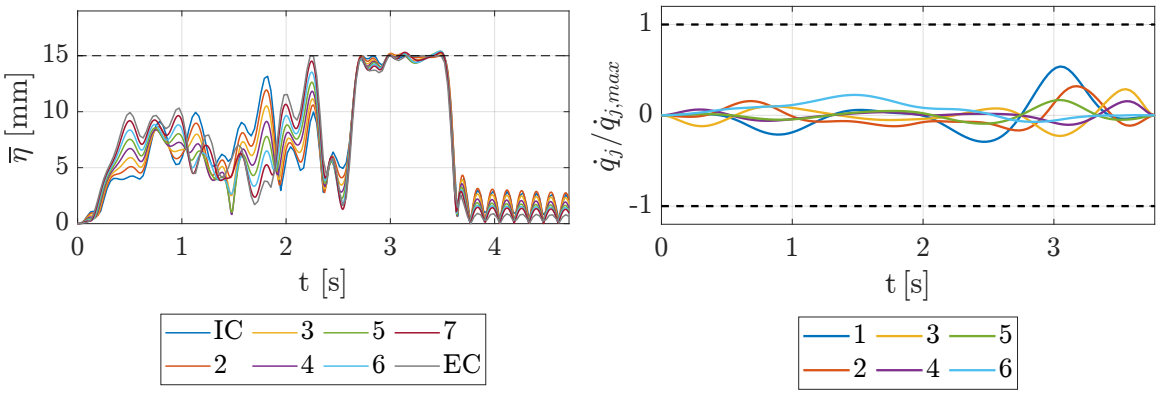
$$\frac{d}{dt}\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} 1 + \frac{C_1^2}{R^2} x_i^2 & \frac{C_1^2}{R^2} x_i y_i \\ \frac{C_1^2}{R^2} x_i y_i & 1 + \frac{C_1^2}{R^2} x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} f_{i,x} \\ f_{i,y} \end{bmatrix} \qquad i = IC, EC \tag{5.8c}$$

where:

$$\begin{aligned} f_{i,x} = &-\frac{C_1^2}{R^2}(\dot{x}_i^2 + \dot{y}_i^2)x_i + (2\dot{\theta}\dot{y}_i + \dot{\theta}^2 x_i + \ddot{\theta} y_i) - \omega_1^2 x_i \left[1 + \alpha_1 (x_i^2 + y_i^2)^{w-1}\right] \\ &- 2\omega_1\zeta_1 \left[\dot{x}_i + \frac{C_1^2}{R^2}(x_i\dot{x}_i + y_i\dot{y}_i)x_i\right] - \ddot{r}_{i,x}\cos(\theta) - \ddot{r}_{i,y}\sin(\theta) - \ddot{r}_{i,z}\frac{C_1}{R}x_i, \\ f_{i,y} = &-\frac{C_1^2}{R^2}(\dot{x}_i^2 + \dot{y}_i^2)y_i + (-2\dot{\theta}\dot{x}_i + \dot{\theta}^2 y_i - \ddot{\theta} x_i) - \omega_1^2 y_i \left[1 + \alpha_1 (x_i^2 + y_i^2)^{w-1}\right] \\ &- 2\omega_1\zeta_1 \left[\dot{y}_i + \frac{C_1^2}{R^2}(x_i\dot{x}_i + y_i\dot{y}_i)y_i\right] + \ddot{r}_{i,x}\sin(\theta) - \ddot{r}_{i,y}\cos(\theta) - \ddot{r}_{i,z}\frac{C_1}{R}y_i. \end{aligned} \tag{5.8d}$$

The way-volumes are imposed in equation (5.7h), where n_v represents the number of spherical way-volumes, with $\rho = 50$ mm. Condition (5.7i) ensures that the computed

path does not assume forbidden vertical positions. Regarding the joint variables constraint of equation (5.7j), the following considerations can be made. Unlike the previous section, where constraint (5.3h) merely served as an indicator that the trajectory had been correctly chosen within the robot workspace, without depending on the solution found by the optimizer; here, constraint (5.7j) is influenced both by the selection of the way-volumes and by the optimizer itself. In particular, it is necessary to define way-volumes that lie within the robot workspace to ensure that the constraints on $\mathbf{q}$ are satisfied at the time instants t_j. However, the optimizer must also enforce this constraint throughout the remaining portion of the computed trajectory. Furthermore, after the optimization process, it is advisable to perform a Jacobian analysis of the manipulator during motion, ensuring that its determinant does not approach zero, thus avoiding configurations of singularity. Regarding the remaining equations of (5.7), the same considerations made in section 5.2.1 apply.
As an example, figure 5.5 presents the optimization results for Trajectory Volumes 2, considering 8 containers and a sloshing height limit of $\overline{\eta}_{lim}$ = 15 mm. Specifically, figure 5.5a illustrates the sloshing-height trends for each container, while figure 5.5b displays the normalized joint velocity profiles of the robot. It can be observed that the joint velocities remain significantly below their limits. This occurs because the constraints that reach saturation in the optimization are those on sloshing heights. Indeed, as shown in figure 5.5a, these values frequently reach $\overline{\eta}_{lim}$.



(a) Sloshing-height trends.

(b) Normalized joint velocities.

Figure 5.5: Trajectory Volumes 2 results.

5.3 Experimental results

This section presents the experimental results, specifically showing the trends of the sloshing heights of the *IC* and *EC* containers for different trajectories and various imposed values of $\overline{\eta}_{lim}$. The predicted sloshing heights from the model will be compared with the experimental ones obtained from the image-processing algorithm. To demonstrate the effectiveness of the optimized motion laws, for each prescribed path, experiments were conducted using also a standard motion law, specifically a modified trapezoidal law in acceleration, shown in figure 5.6. The standard motion law was defined by imposing the same duration of the corresponding optimized motion law.

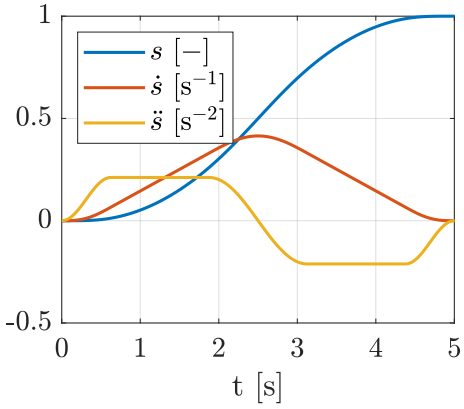


Figure 5.6: Modified trapezoidal motion-law.

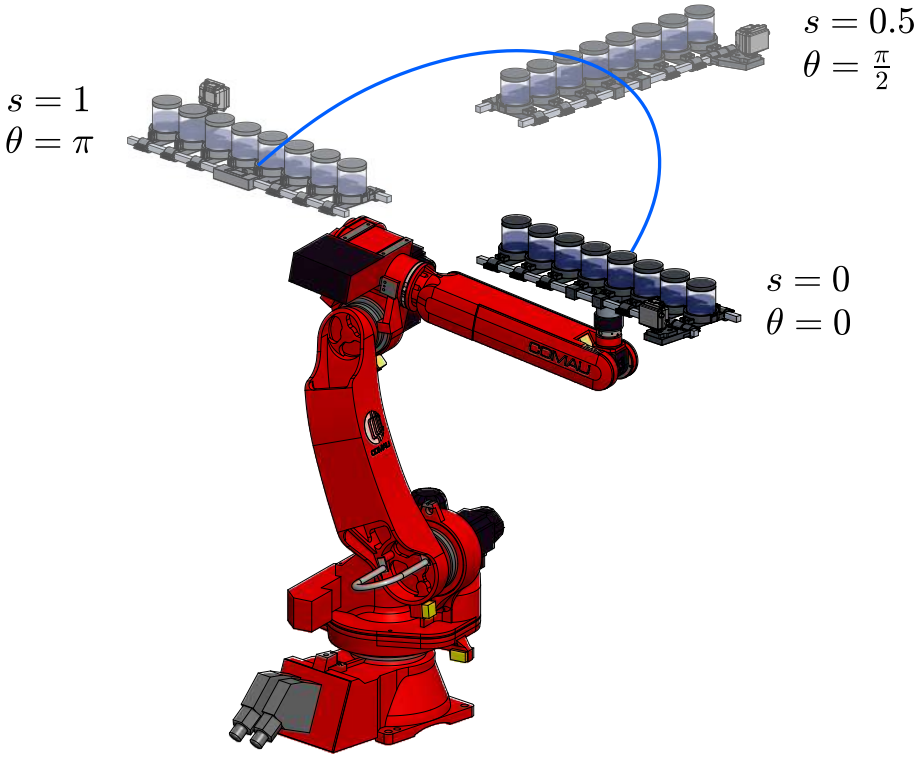
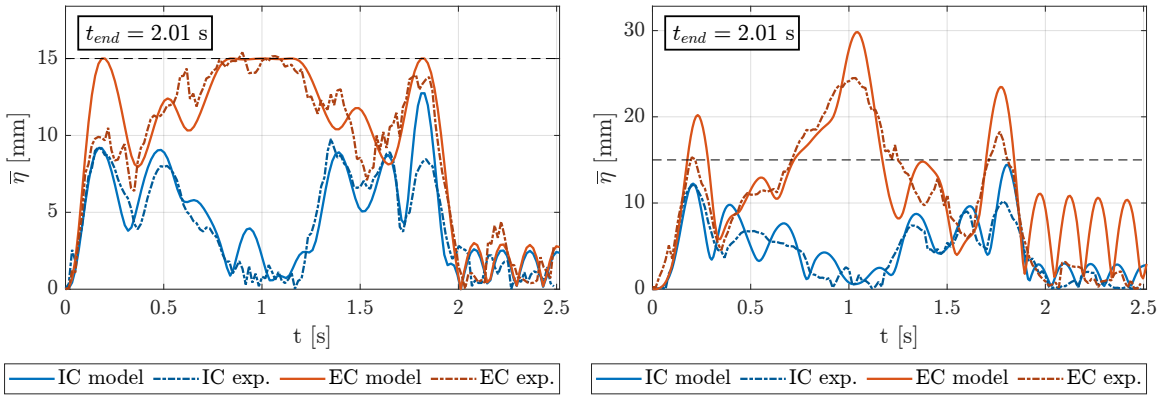


Figure 5.7: Trajectory 1A.

5.3.1 Trajectory 1A

The results for the first considered trajectory, called *Trajectory 1A* in accordance with what was presented in the previous chapters, are reported below. The trajectory is shown in figure 5.7, where the tray undergoes a total rotation of π. Figure 5.8 shows the trends of the sloshing heights with the optimized motion law considering $\overline{\eta}_{lim}$ = 15 mm (figure 5.8a) and trapezoidal law with the same duration (figure 5.8b); whereas in figure 5.9, the same trends are shown for the $\overline{\eta}_{lim}$ = 25 mm case. In the aforementioned figures, the continuous blue line represents the trend of $\overline{\eta}$ for the IC container predicted by the mass-spring-damper NL model whereas the dashed blue line indicates the experimentally obtained $\overline{\eta}$ for the same container. It can be observed that the two trends are very similar, demonstrating the accuracy of the NL model in predicting the sloshing height. Similarly, the red curves refer to the EC container, with the solid line representing the NL-model prediction and the dashed line showing the

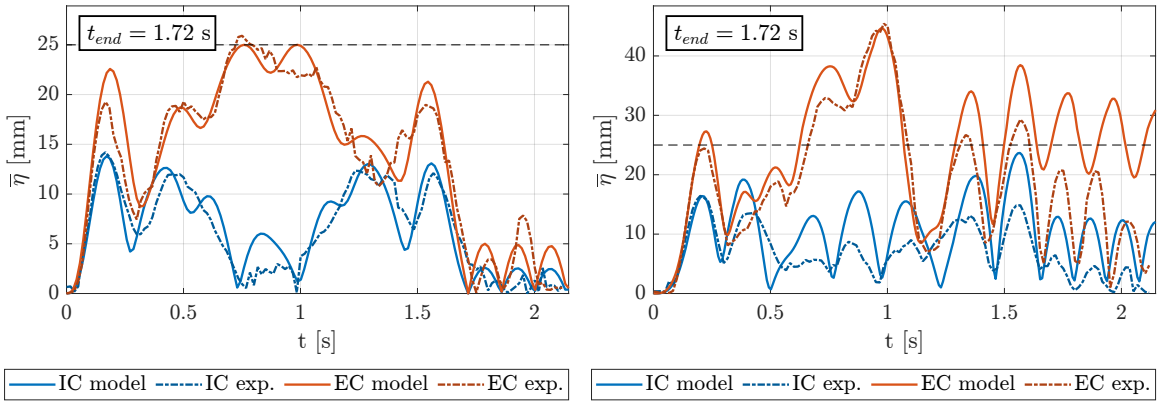
experimentally obtained trends. It is evident that using the optimized motion laws ensures that the sloshing heights do not exceed the imposed limit, within a certain tolerance of a few millimeters (due to the fact that the liquid model is approximate, and some errors may occur in sloshing-height detection by the GoPro cameras). On the other hand, looking at the modified trapezoidal motion law graphs, it is noticeable that, for the same t_{end}, the sloshing heights far exceed the limit, nearly doubling it. To



(a) Optimized motion law.

(b) Non-optimized motion-law with the same t_{end}.

Figure 5.8: Sloshing-height comparison for Trajectory 1A with $\overline{\eta}_{lim}$ = 15 mm in the optimized case.



(a) Optimized motion law.

(b) Non-optimized motion-law with the same t_{end}.

Figure 5.9: Sloshing-height comparison for Trajectory 1A with $\overline{\eta}_{lim}$ = 25 mm in the optimized case.

further highlight the effectiveness of the optimized motion law compared to the non-optimized one, the percentage difference between the maximum peak of $\overline{\eta}$ obtained with the non-optimized motion law and the maximum peak of $\overline{\eta}$ obtained with the optimized motion law with the same t_{end} is defined:

$$\gamma_{\overline{\eta},\%} = 100\frac{\overline{\eta}_{max,Nopt} - \overline{\eta}_{max,Opt}}{\overline{\eta}_{max,Nopt}}. \tag{5.9}$$

In table 5.1, the four ratios $\gamma_{\overline{\eta},\%}$ obtained by considering IC, EC, $t_{end} = 2.01$ s, and $t_{end} = 1.72$ s are reported.

Container	$t_{end} = 2.01$ s	$t_{end} = 1.72$ s
IC	$\gamma_{\overline{\eta},\%} = 19.0\%$	$\gamma_{\overline{\eta},\%} = 13.9\%$
EC	$\gamma_{\overline{\eta},\%} = 37.4\%$	$\gamma_{\overline{\eta},\%} = 43.0\%$

Table 5.1: Ratios between sloshing-height peaks obtained with non-optimized and optimized motion law for Trajectory 1A.

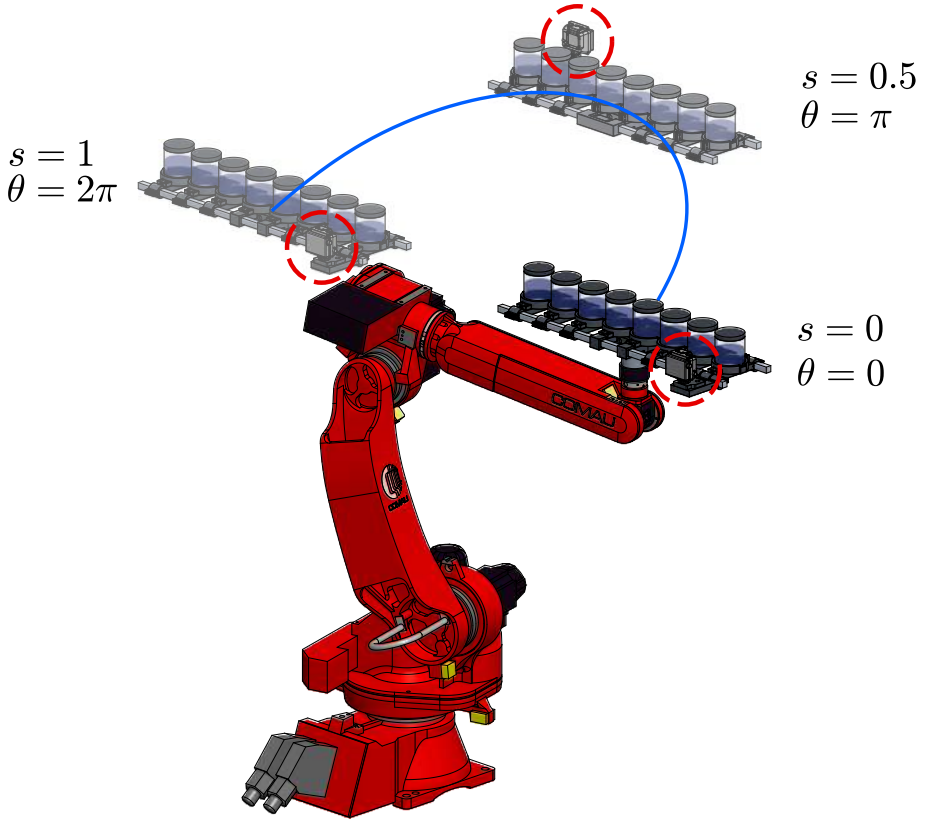
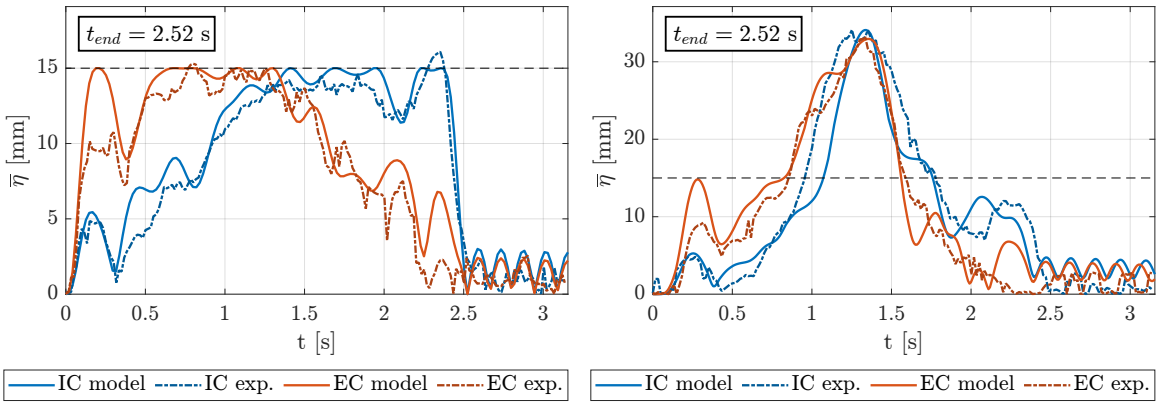


Figure 5.10: Trajectory 1B.

5.3.2 Trajectory 1B

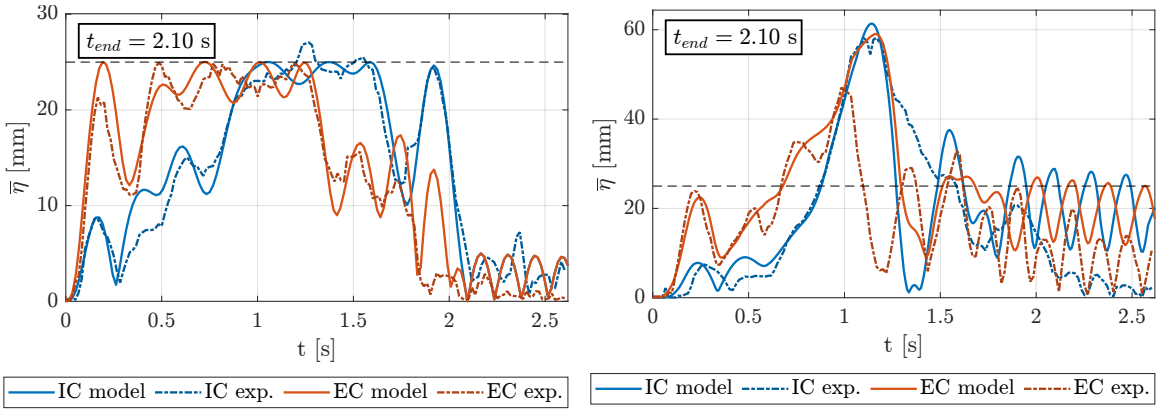
The results for the second trajectory, called *Trajectory 1B* in accordance with what was presented in the previous chapters, are reported below. The trajectory is shown in figure 5.10, the tray undergoes a total rotation of 2π. The change of the tray orientation in the three configurations shown in the figure can be better appreciated by noting the position of the GoPro, circled in red. Figure 5.11 shows the trends of the sloshing heights with the optimized motion law considering $\overline{\eta}_{lim}$ = 15 mm (figure 5.11a) and the trapezoidal law with the same duration (figure 5.11b); in figure 5.12, the same trends are shown for the case with $\overline{\eta}_{lim}$ = 25 mm. Again, it can be observed that the optimized motion laws allow the imposed limits to be respected within a small tolerance. Regarding the non-optimized cases, they show maximum sloshing heights even higher than those observed in *Trajectory 1A*, further highlighting the benefits of using the optimized motion laws. As in previous section, the percentage differences between the maximum peak of $\overline{\eta}$ obtained with the non-optimized motion law and the the maximum peak of $\overline{\eta}$ obtained with the optimized motion law with the same t_{end}, defined by equation 5.9, are reported in table 5.2. The ratios presented in the table, for the cases with t_{end} = 2.10 s, would actually be even higher. The peaks corresponding to the experimental trends of $\overline{\eta}$, for both the *IC* and *EC* containers in figure 5.12b, would indeed be higher. In fact, by analyzing the videos recorded with the GoPro cameras, it can be observed that, at the instant of the maximum peak (shown in figures 5.13), the liquid impacts against the lid sealing the container. As a result, the $\overline{\eta}$ value computed from the post-processing of the videos is limited to the height of the lid.



(a) Optimized motion law.

(b) Non-optimized motion-law with the same t_{end}.

Figure 5.11: Sloshing-height comparison for Trajectory 1B with $\overline{\eta}_{lim} = 15$ mm in the optimized case.



(a) Optimized motion law.

(b) Non-optimized motion-law with the same t_{end}.

Figure 5.12: Sloshing-height comparison for Trajectory 1B with $\overline{\eta}_{lim} = 25$ mm in the optimized case.

Container	$t_{end} = 2.52$ s	$t_{end} = 2.10$ s
IC	$\gamma_{\overline{\eta},\%} = 52.8\%$	$\gamma_{\overline{\eta},\%} = 53.4\%$
EC	$\gamma_{\overline{\eta},\%} = 54.1\%$	$\gamma_{\overline{\eta},\%} = 47.1\%$

Table 5.2: Ratios between sloshing-height peaks obtained with non-optimized and optimized motion law for Trajectory 1B.

(a) IC. (b) EC.

Figure 5.13: Instantaneous frame at the moment of maximum $\overline{\eta}$ for Trajectory 1B with a non-optimized motion law of duration $t_{end} = 2.10$ s.

5.3.3 Trajectory 2

The results for the third trajectory, called *Trajectory 2* in accordance with what was presented in the previous chapters, are reported below. The trajectory is shown in figure 5.14, with the tray undergoing a total rotation of π. The change of the tray orientation in the two configurations shown in the figure can be better appreciated by noting the position of the GoPro on the tray. The sloshing-height trends obtained with the optimized motion law with $\overline{\eta}_{lim} = 15$ mm are shown in figure 5.15, while the sloshing-height trends obtained with the optimized motion law with $\overline{\eta}_{lim} = 25$ mm are shown in figure 5.16. Before presenting the trends obtained with the non-optimized motion law, it is necessary to make some considerations regarding figures 5.15 and 5.16. Unlike the trends of $\overline{\eta}$ with optimized motion laws for Trajectory 1A and Trajectory 1B, in this case, it can be observed that the experimental trends exceed the value of $\overline{\eta}_{lim}$ by a few millimeters at certain points. Specifically, for the container IC, in the case shown in figure 5.15, the maximum recorded value of $\overline{\eta}$ is 20.6 mm, compared to an expected $\overline{\eta}_{lim} = 15$ mm. Similarly, in figure 5.16, the same container IC registers a maximum value of $\overline{\eta}$ of 36.0 mm, against an expected $\overline{\eta}_{lim} = 25$ mm. These errors, significantly larger than those recorded for the previous two trajectories, are due to the shape of the assigned path for this trajectory. In fact, as shown in figure 5.14, the imposed path presents sharp turns, almost at right angles. Traversing these curves causes instantaneous centripetal accelerations, which act similarly to impacts on the liquid. The optimizer still attempts to manage these abrupt changes in direction, and indeed, the computational trends (represented by the continuous lines) in figures 5.15 and 5.16

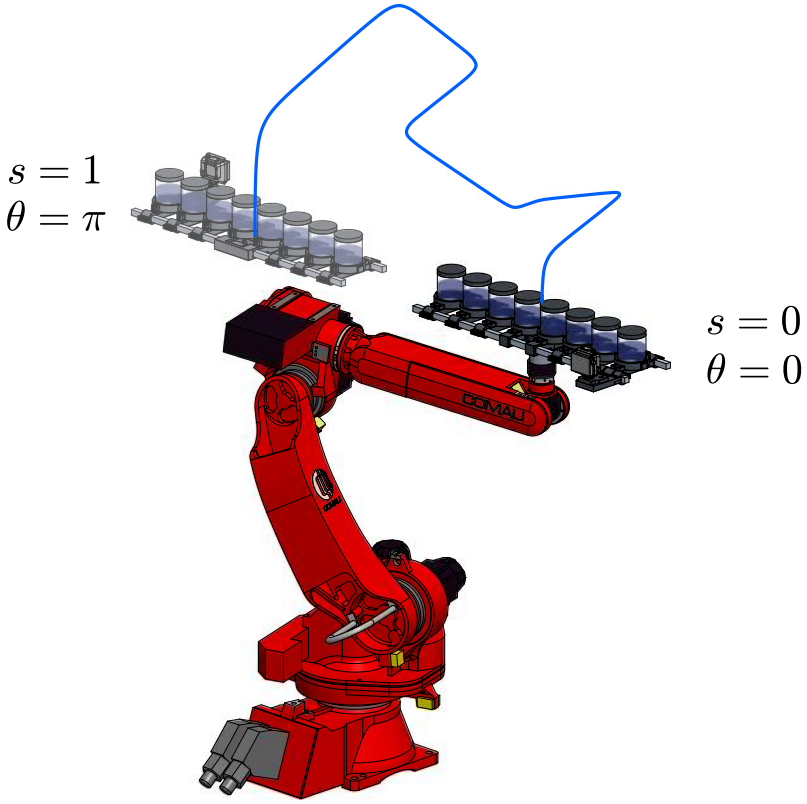


Figure 5.14: Trajectory 2.

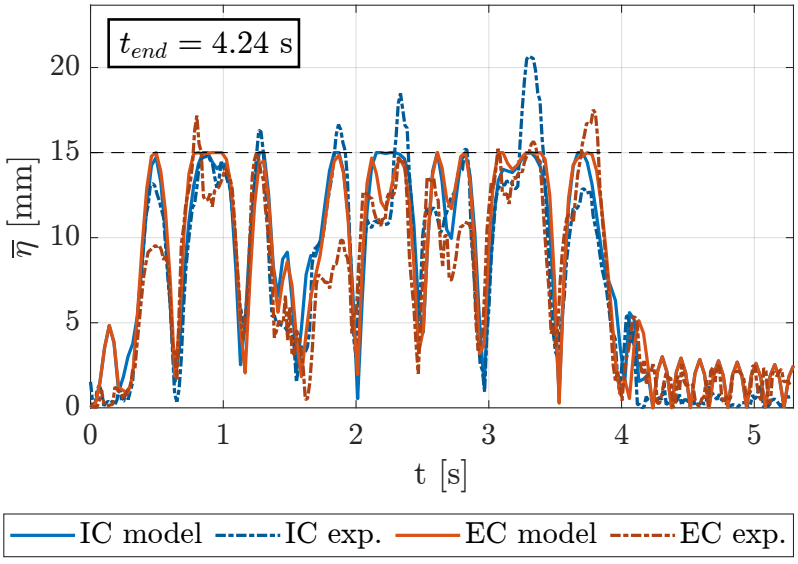


Figure 5.15: Trajectory 2 sloshing height: optimized motion-law with $\overline{\eta}_{lim}$ = 15 mm.

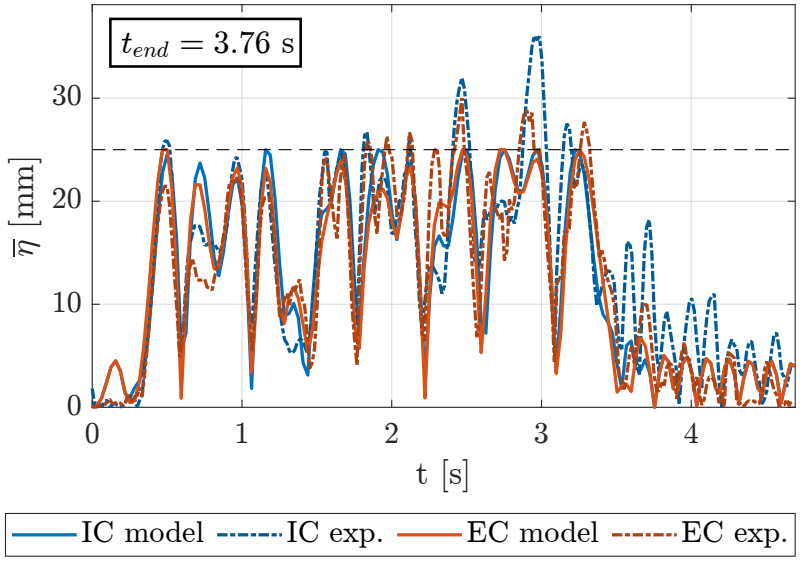


Figure 5.16: Trajectory 2 sloshing height: optimized motion-law with $\overline{\eta}_{lim} = 25$ mm.

remain below the imposed limit. The issue arises in the experimental trends, which remain undetectable to the optimizer, because the repeated impacts caused by these directional changes establish a motion regime in the liquid that the NL model is unable to capture. As explained in section 2.1, three different motion regimes can occur for a liquid inside a cylindrical container. The first, caused by low excitations, results in a planar liquid surface (figure 2.2a) and is described by the L model. The second, occurring under intermediate excitations, results in a liquid surface described by a Bessel function (figure 2.2b) and is modeled by the NL model. The third regime is characterized by strong nonlinearities, causing instantaneous peaks in the liquid surface. For these optimizations, the NL model is being used, which allows modeling of the second motion regime but not the third. By analyzing the GoPro footage, particularly for the optimizations with $\overline{\eta}_{lim} = 25$ mm (figure 5.16), it can be observed that, at certain moments, the liquid surface exhibits a shape that falls into the third regime. An example of this observation is shown in figure 5.17. In conclusion, the peaks exceeding $\overline{\eta}_{lim}$ are not caused by the optimizer but rather by the limitations of the NL model, which fails to capture strong nonlinearities in the liquid. However, this does not pose an issue for our applications. Indeed, the algorithms developed in this thesis are specifically designed to optimize simple pick-and-place motions, such as those in Trajectory 1A and Trajectory 1B, where the optimizer shows its efficacy, as demonstrated in sections 5.3.1 and 5.3.2. Trajectory 2 has been used throughout the thesis as an extreme case, presenting a highly challenging scenario for the optimizer. However, it does not represent a realistic industrial pick-and-place motion. Despite this, the optimizer has consistently provided acceptable results even for this trajectory. Except in cases where the NL model fails, the optimizer still delivers highly suitable solutions compared to non-optimized cases. Based on this observation, the following section presents the results obtained using a modified trapezoidal motion law, which is non-optimized. Unlike in sections 5.3.1 and 5.3.2, results are not reported for two non-optimized motion laws with the same t_{end} of the optimized cases for $\overline{\eta}_{lim} = 15$ mm and $\overline{\eta}_{lim} = 25$ mm. For this trajectory, it was not feasible to use a trapezoidal motion law with $t_{end} = 4.24$ s or $t_{end} = 3.76$ s since executing trajectories with these values would result in exceeding the robot joint acceleration limits. The sharp turns in Trajectory 2, when executed

Figure 5.17: Instant peak of the liquid: third regime of motion; *EC* container during optimized motion-law with $t_{end} = 3.76$ s.

at excessive velocity, generate high centripetal accelerations. To address this, a single non-optimized motion law with an extended duration of $t_{end} = 6$ s was used, this t_{end} is 29.3% longer than $t_{end} = 4.24$ s (figure 5.15) and 37.3% longer than $t_{end} = 3.76$ s (figure 5.16). The sloshing heights obtained with this motion law are shown in figure 5.18. Despite the significantly increased t_{end}, the sloshing heights are very high. The graph reveals that after the initial peak, the experimental trends diverge from the predicted ones. As previously explained in relation to figure 5.12b, this occurs because the liquid impacts the container lid, altering its subsequent dynamics. Similarly to what was done in the previous sections, table 5.3 reports the percentage differences between the maximum sloshing heights recorded using a non-optimized motion-law (figure 5.18) and the maximum sloshing heights obtained with the optimized motion-law (figures 5.15 and 5.16), defined by equation 5.9. Unlike previous sections, as previously mentioned, the two $\overline{\eta}$ values used for the ratios in table 5.3 were not obtained using motion laws with equal durations, as shown in the first row of the table, where the upper number represents the t_{end} of the non-optimized motion law, while the lower number represents the t_{end} of the optimized motion law. In conclusion, in this section it was shown that, for this trajectory, the optimizer was not able to produce a motion law that perfectly complied with the constraints. However, by comparing the results obtained

Container	$t_{end,Nopt} = 6.00$ s $t_{end,Opt} = 4.24$ s	$t_{end,Nopt} = 6.00$ s $t_{end,Opt} = 3.76$ s
IC	$\gamma_{\overline{\eta},\%} = 60.2\%$	$\gamma_{\overline{\eta},\%} = 30.5\%$
EC	$\gamma_{\overline{\eta},\%} = 71.2\%$	$\gamma_{\overline{\eta},\%} = 50.7\%$

Table 5.3: Ratios between sloshing-height peaks obtained with non-optimized and optimized motion law for Trajectory 2.

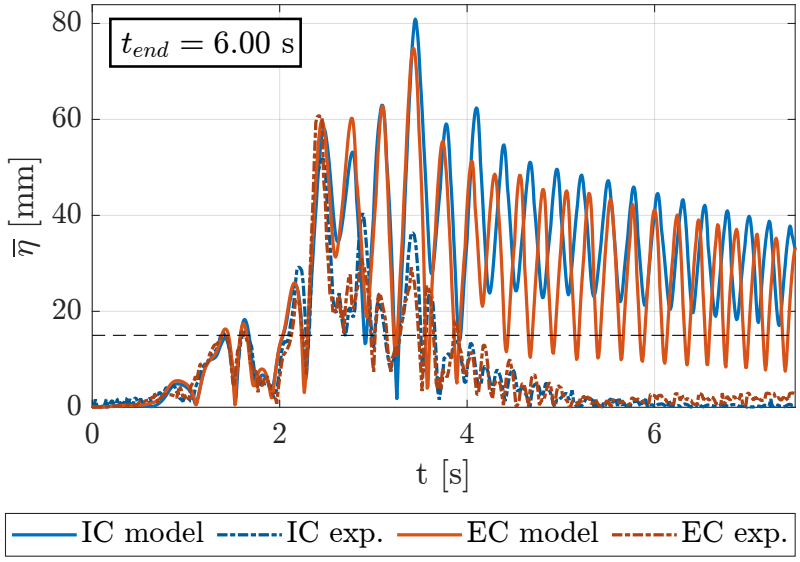


Figure 5.18: Non-optimized motion-law with $t_{end} = 6$ s.

with a non-optimized motion law of even longer duration, it can be observed that the optimizer still achieves a significant reduction in $\overline{\eta}$, as well as in t_{end}.

5.3.4 Trajectory Volumes 2

This section presents the results obtained from point-to-point optimizations with assigned way-volumes. Optimizations with assigned waypoints have not been implemented, since, as discussed in chapter 4, they are not advantageous in certain cases. Unlike the previous sections, the results will not be compared with those obtained using non-optimized motion laws. The purpose of the previous sections was to highlight the benefits of optimizations over a non-optimized motion. Instead, this section aims to emphasize the advantages of point-to-point optimizations over those with a predefined path, demonstrating that they can, within certain limits, overcome the limitations observed in the results of the experiments with assigned path optimizations. Specifically, as shown in section 5.3.3, the maximum limits imposed on the sloshing height were exceeded at certain brief moments during the execution of Trajectory 2. This was due to the complex shape of the path, which induced strong motion nonlinearities in the liquid that could not be accurately represented by the NL model. For this reason, in this section, only the results for *Trajectory Volumes 2* are reported (in accordance with the nomenclature introduced in chapter 4), for completeness, the results for *Trajectory Volumes 1A* and *1B* are provided in appendix C. This trajectory is defined based on way-volumes placed along *Trajectory 2*, which is the trajectory where issues were identified. As observed in section 4.3.3, *Trajectory Volumes 2* exhibits significantly wider curves compared to *Trajectory 2*. The presence of sharp curves was the primary factor responsible for the motion nonlinearities in the liquid, leading to model failure. Thus, this section aims to highlight that not only the computed trajectories have a shorter duration than those with a predefined path, but also that the newly smoothed paths do not introduce nonlinearities that the NL model is unable to handle. Consequently, these new trajectories ensure better compliance with the maximum sloshing-height constraints. Figure 5.19 illustrates the trajectory, where the tray undergoes a total rotation of π. It is evident that the trajectory features much wider curves com-

pared to the one depicted in figure 5.14. The sloshing-height trends obtained with the

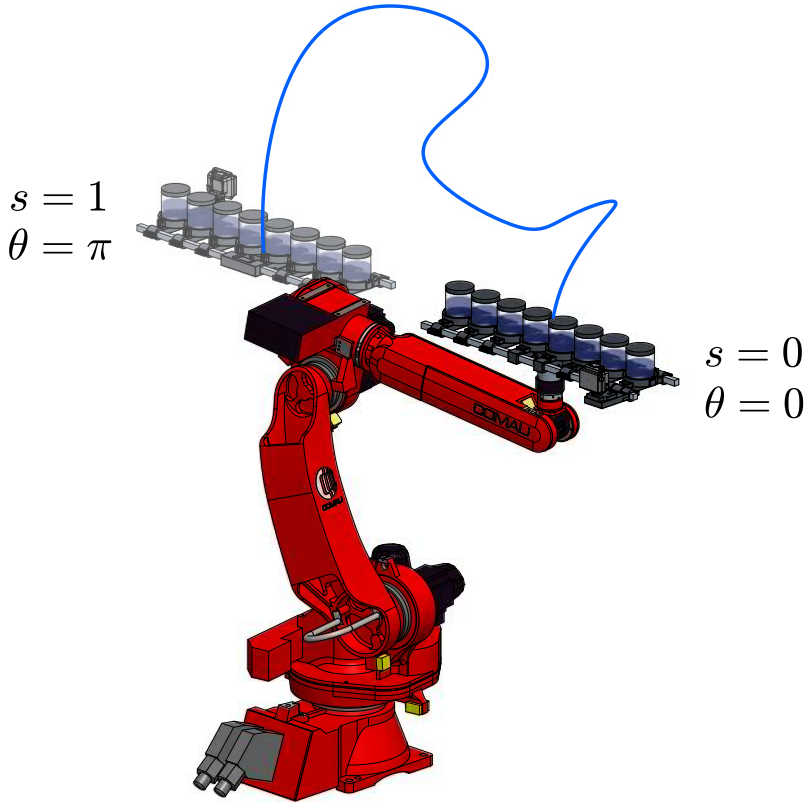


Figure 5.19: Trajectory Volumes 2.

optimized motion law with $\overline{\eta}_{lim}$ = 15 mm are shown in figure 5.20, while the sloshing-height trends obtained with the optimized motion law with $\overline{\eta}_{lim}$ = 25 mm are shown in figure 5.21. Comparing the results with those obtained in the previous section, it is evident that for $\overline{\eta}_{lim}$ = 15 mm, the point-to-point optimization with assigned way-volumes (figure 5.20) results in a lower t_{end} compared to the optimization with a predefined path (figure 5.15), while also correctly satisfying the constraints on the maximum sloshing heights. This is due to the fact that, as previously explained, the trajectory computed through point-to-point optimization does not include curves that induce motion regimes beyond the modeling capabilities of the NL model. In conclusion, for the analyzed case, this optimization approach provides a significant advantage both in terms of trajectory duration and compliance with $\overline{\eta}_{lim}$. Regarding the results obtained for the optimization with $\overline{\eta}_{lim}$ = 25 mm (figure 5.21), it can be observed that, similarly to the previous case, the trajectory is not only shorter than that obtained with the predefined path optimization (figure 5.16), but also results in lower sloshing-height peaks, ensuring better compliance with the constraints on $\overline{\eta}$. However, unlike the case with $\overline{\eta}_{lim}$ = 15 mm, at the end of the trajectory, the imposed limit of $\overline{\eta}_{lim}$ = 25 mm is slightly exceeded, with recorded values of $\overline{\eta}_{max,IC}$ = 28.1 mm and $\overline{\eta}_{max,EC}$ = 29.5 mm. An even more critical issue is the sloshing-height trend of container IC after the end of the motion, which significantly exceeds the imposed limit of $0.2\overline{\eta}_{lim}$ = 5 mm. As discussed in section 5.3.3, this discrepancy between simulated and experimental trends is attributed to the nonlinearities of the liquid surface, which give rise to a third-type motion regime. To support this statement, figure 5.22 presents an instantaneous frame of container IC at the end of the motion, where a peak is visible (even though of lower magnitude than the one in figure 5.17), contributing to the larger residual oscillations

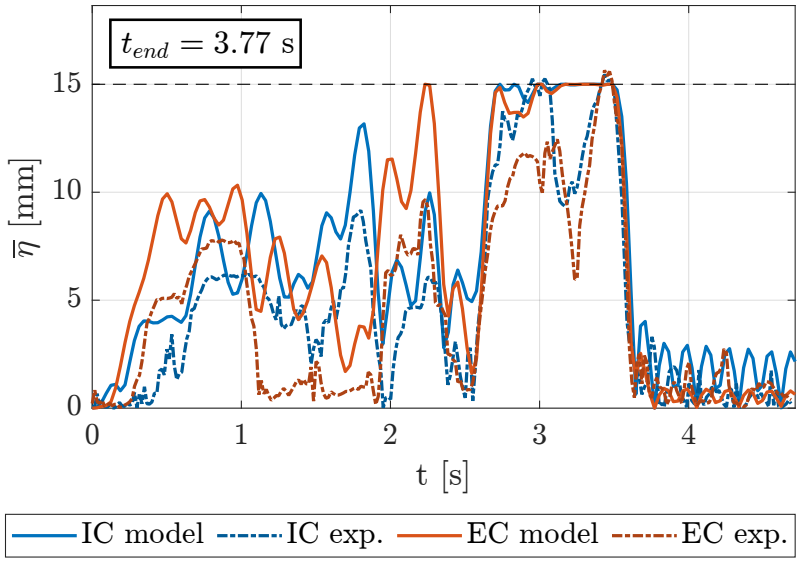


Figure 5.20: Trajectory Volumes 2 sloshing height: optimized motion-law with $\overline{\eta}_{lim}$ = 15 mm.

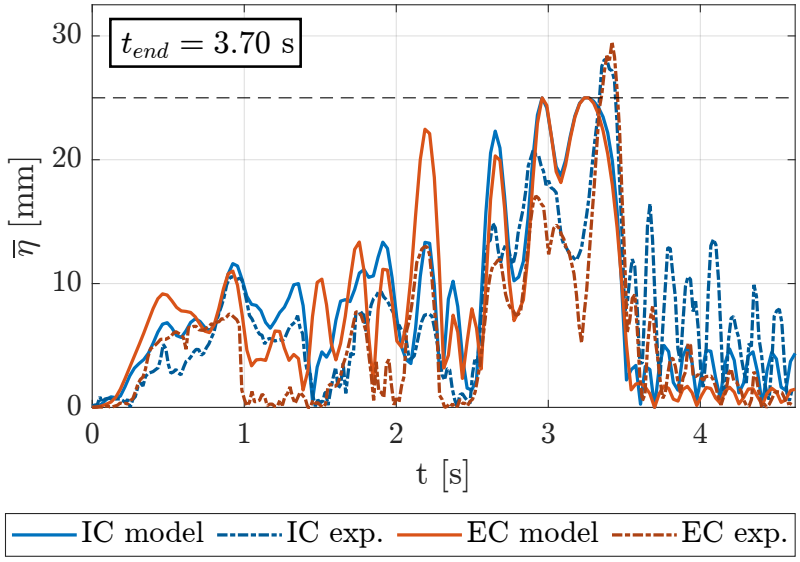


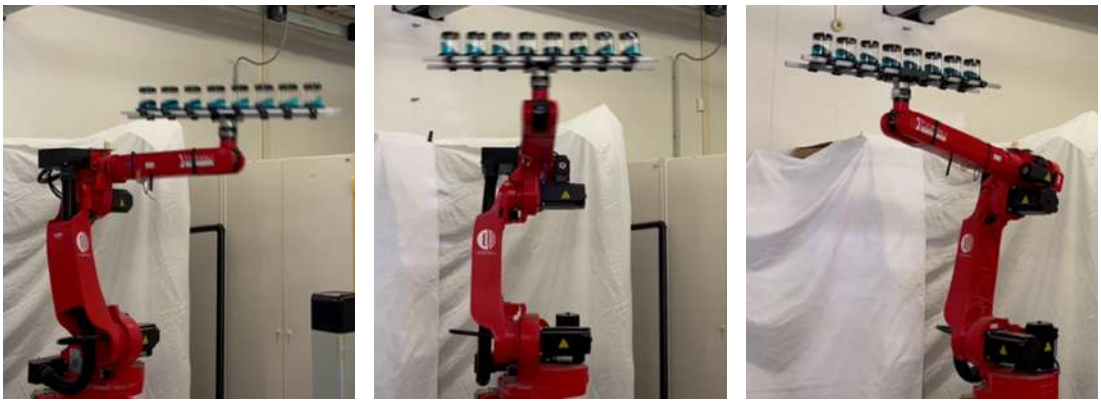
Figure 5.21: Trajectory Volumes 2 sloshing height: optimized motion-law with $\overline{\eta}_{lim}$ = 25 mm.

Figure 5.22: Instant peak of the liquid, container *IC* at the end of *Trajectory Volumes 2* wuth $\overline{\eta}_{lim} = 25$ mm.

after t_{end}. In conclusion, for this second case, where $\overline{\eta}_{lim} = 25$ mm was imposed, the issues discussed in the previous section were still observed, but with lower sloshing-height peaks and for shorter trajectory durations, ultimately confirming the advantage of the point-to-point optimization with assigned way-volumes in this scenario as well.

5.3.5 Trajectory instant frame

Figure 5.23 shows frames from videos recorded during the execution of Trajectory 1A, 1B, and 2. The free surfaces of the liquid in the various containers can be observed. As expected from the sloshing-height trends presented in this thesis, for Trajectory 1A (figure 5.23a), the highest sloshing heights are recorded in the outer container (the one on the right). In Trajectory 1B (figure 5.23b), the maximum sloshing heights appear at both the external containers, whereas for Trajectory 2 (figure 5.23c), the sloshing heights in the various containers are approximately the same in that frame. This is because, for this trajectory, translational motion is predominant over rotational motion.



(a) Trajectory 1A. (b) Trajectory 1B. (c) Trajectory 2.

Figure 5.23: Instantaneous frames of robot trajectories.

Chapter 6

Conclusions

In this thesis, several optimization approaches were developed for time-optimal trajectory planning for anti-sloshing robotic manipulation of multiple liquid-filled containers. The objective was to determine the shortest trajectory to simultaneously transport multiple liquid-filled containers using a robotic manipulator while ensuring that the sloshing height of the liquid in each container remained within a specified threshold. Four-dimensional trajectories were considered, including 3-dimensional translations and a 1-dimensional rotation about a vertical direction. After analyzing the state of the art, a mass-spring-damper model was selected to represent the sloshing dynamics within the containers, adopting constrained optimization as the technique to mitigate it.

Two families of optimizations were developed. In the first family, presented in chapter 3, the optimal motion laws were determined to follow pre-assigned paths. This chapter included numerous analyses aimed at identifying the best strategy for performing the optimizations. It was concluded that, with multiple containers mounted in line on a rigid support, it is sufficient to impose a constraint on the sloshing height of the two outer containers, indicated as IC and EC. This conclusion enabled the development of algorithms independent of the number of containers, thus ensuring high application flexibility. The second family of optimizations, analyzed in chapter 4, focused on point-to-point motions and was aimed at determining both the optimal motion law and the optimal path. In these cases, the path was not predefined; instead, only waypoints or way-volumes were assigned. These optimizations were implemented to identify trajectories even faster than in the first case. By allowing the path to vary in addition to the motion law, the optimizer faced fewer constraints, and consequently, it was able to find better solutions. For point-to-point optimizations with a single assigned waypoint, a reduction in trajectory duration was observed. However, as the number of waypoints increased, the advantage of this approach decreased, and in some cases, it resulted in worse outcomes compared to the optimization with a predefined path. This issue was resolved by adopting point-to-point optimizations with assigned way-volumes, which led to a significant improvement in trajectory duration, even in cases where the approach with waypoints was ineffective. Chapter 5 described the experimental campaign aimed at validating the algorithms presented in the previous chapters. A support structure capable of holding 8 containers was designed, mounted on the EE of a 6-DOF industrial serial robot. Using a camera system fixed on the tray hosting the containers, the liquid behavior in the outer containers during motion was recorded, and the sloshing-height experimental trends were measured through post-processing of

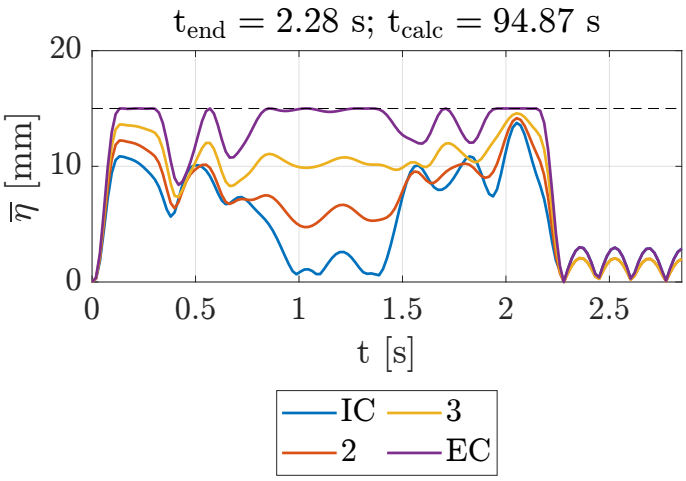
the videos. For the optimizations with a predefined path, experiments were conducted on three 4-dimensional trajectories. The first two trajectories represented simple pick-and-place movements, while the third was a highly complex motion, characterized by numerous curves, in order to represent a limiting application case. The results of the first two trajectories fully confirmed the simulations: the optimized motion laws not only ensured sloshing heights within the imposed limit, but also reduced these heights to a maximum of 54% compared to those obtained with a non-optimized motion law, while keeping the path and trajectory duration unchanged. Regarding the third trajectory, the optimized motion law produced significantly lower sloshing heights compared to those obtained with an even slower non-optimized one, with differences exceeding 70%. From this perspective, the optimization proved successful even for the third trajectory. However, some minor issues arose in complying with the constraint on the maximum sloshing-height value: although the optimized trajectory led to a significant reduction of sloshing heights, there were moments when peaks exceeded the imposed limit. This was due to the fact that, because of the strong nonlinearities of the motion resulting from the complex path shape, the liquid assumed a dynamic regime that the employed mass-spring-damper model was unable to adequately represent. Therefore, it can be concluded that, even for the third trajectory, the optimization yielded positive results, ensuring significant reductions in sloshing heights, and in cases where the limit was exceeded, the issue was attributed to the adopted liquid model rather than to the optimizer. Actually, this issue does not represent an obstacle for the applications for which the optimizations were developed, namely, for simple pick-and-place movements in industrial settings, as represented by the first two trajectories, where both the model and the optimizer provided highly acceptable results. These issues encountered in the experiments with predefined path optimizations are significantly minor, and sometimes entirely resolved, when point-to-point optimizations with assigned way-volumes are performed. In chapter 5, the results of these optimizations were presented for cases where the predefined-path optimization failed. A significant improvement in constraint satisfaction was observed, with lower maximum sloshing heights recorded as a result of trajectories with even shorter durations. Point-to-point optimization with assigned way-volumes thus emerged as the best option from all perspectives and is therefore the recommended optimization strategy to achieve the best results in terms of trajectory duration and constraint satisfaction.
Regarding future developments, a further analysis could focus on techniques for global optimization, aimed at determining even faster trajectories. As explained throughout the thesis, the main obstacle in adopting these techniques lies in the resulting computational time, as the studied problems have a high computational complexity. An interesting study could involve combining the constrained optimizations presented with input-shaping techniques. Indeed, as observed, when dynamics become high and the paths are complex, the used model fails in certain points, as it cannot accurately model the liquid when the motion conditions become highly nonlinear. Applying a filtering technique, such as input shaping, to the motion law obtained through optimization could help compensate for the model errors by eliminating frequency components close to the sloshing modes. A final interesting aspect to explore and integrate into the optimization is tilting. Integrating tilting compensatory motions for very simple trajectories could bring benefits without excessively increasing the computational time of the problem.

Appendices

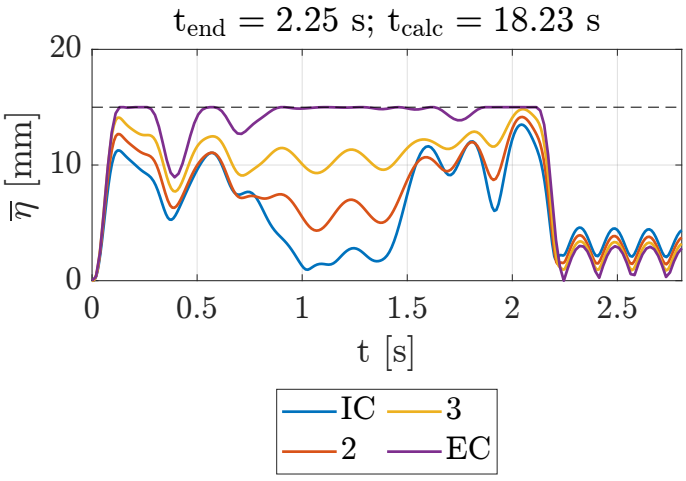
Appendix A

Additional sloshing-height trends

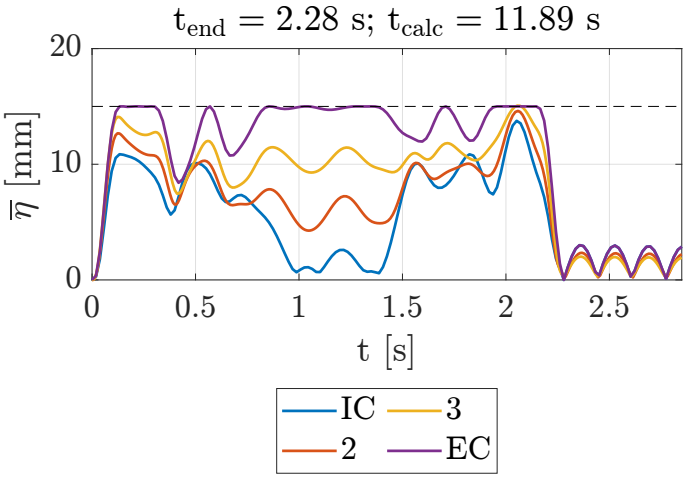
The trends of the sloshing heights for Trajectories 1A, 1B, and 2 are presented below. The trends for each container are shown considering the configurations with 4 and 6 containers. Each figure presents the optimization results by constraining the sloshing height of: each container, only EC, both IC and EC.



(a) $\overline{\eta}_{lim}$ on all.

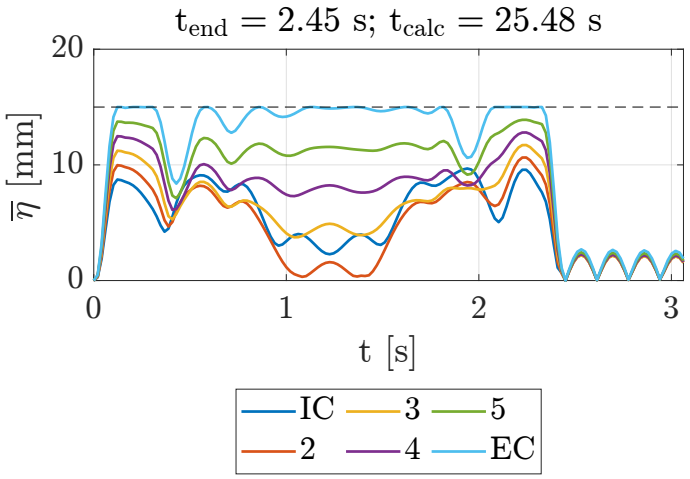


(b) $\overline{\eta}_{lim}$ on EC.

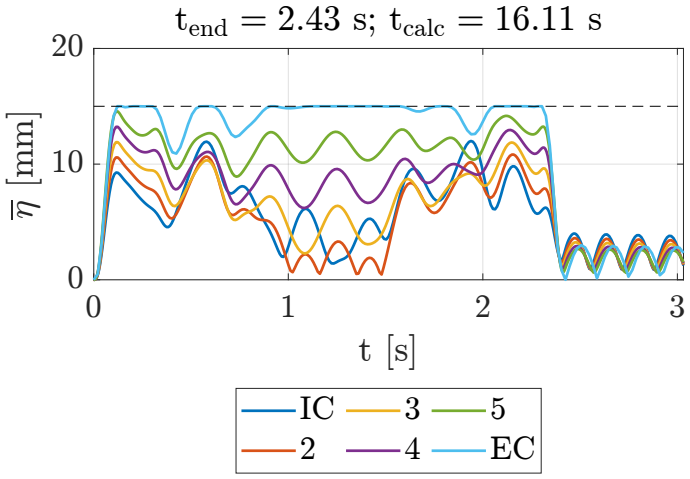


(c) $\overline{\eta}_{lim}$ on IC and EC.

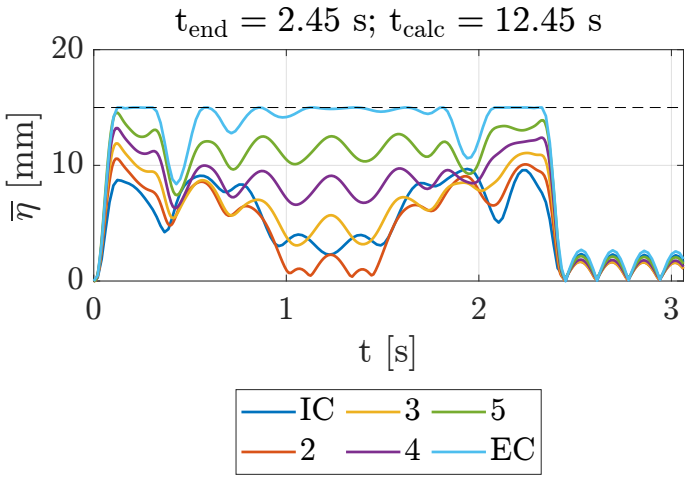
Figure A.1: Trajectory 1A, 4 containers.



(a) $\overline{\eta}_{lim}$ on all.

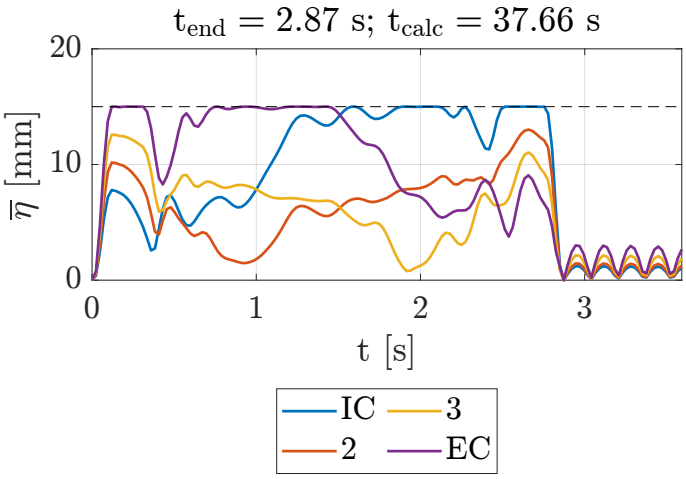


(b) $\overline{\eta}_{lim}$ on EC.

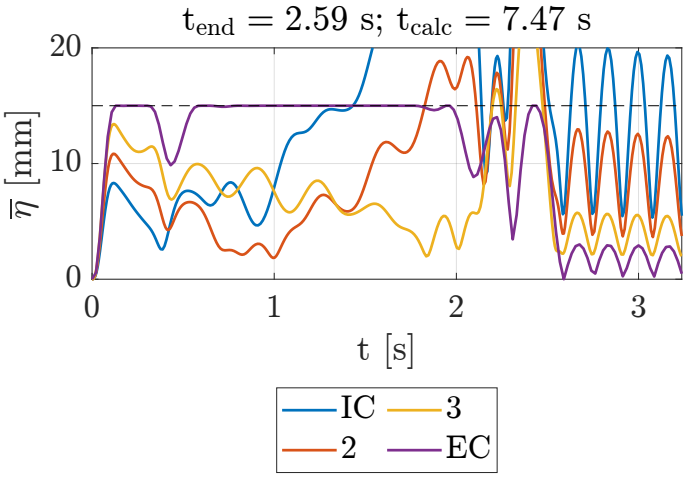


(c) $\overline{\eta}_{lim}$ on IC and EC.

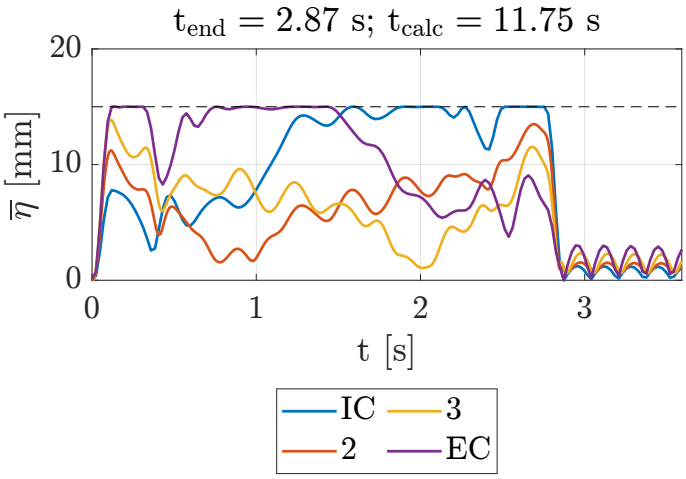
Figure A.2: Trajectory 1A, 6 containers.



(a) $\overline{\eta}_{lim}$ on all.

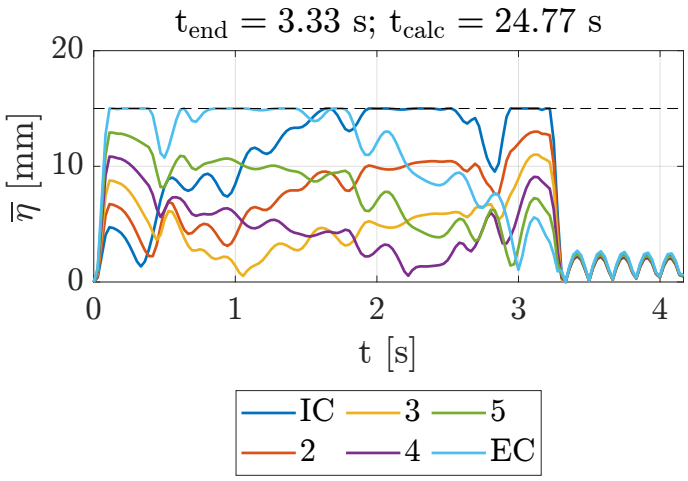


(b) $\overline{\eta}_{lim}$ on EC.

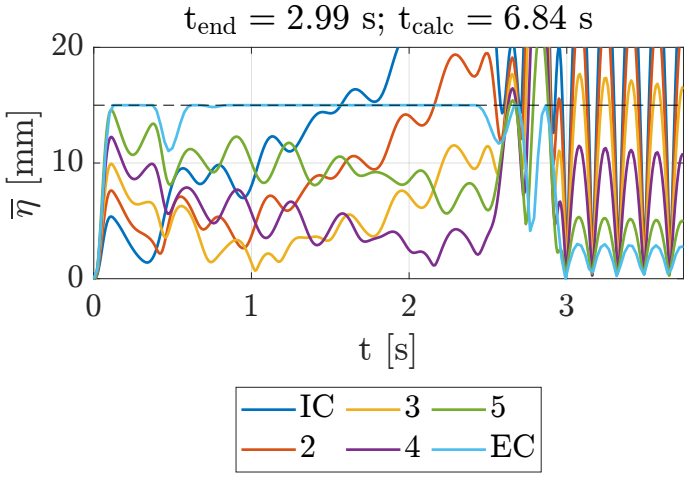


(c) $\overline{\eta}_{lim}$ on IC and EC.

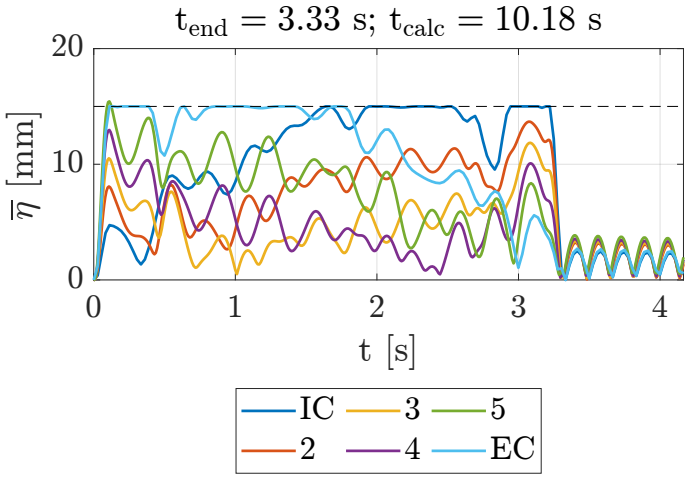
Figure A.3: Trajectory 1B, 4 containers.



(a) $\overline{\eta}_{lim}$ on all.

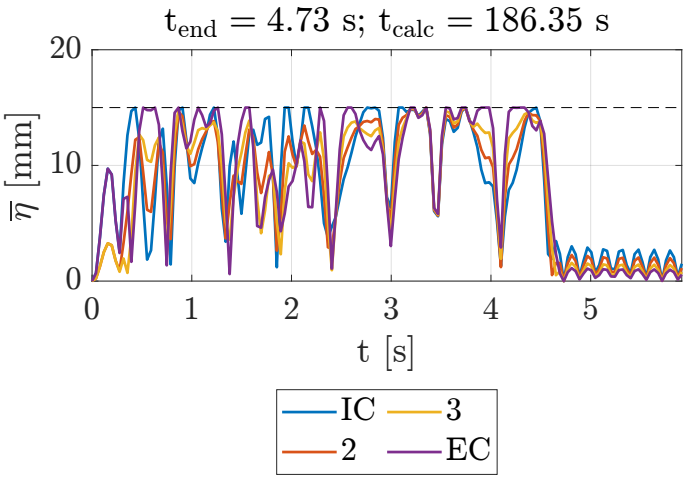


(b) $\overline{\eta}_{lim}$ on EC.

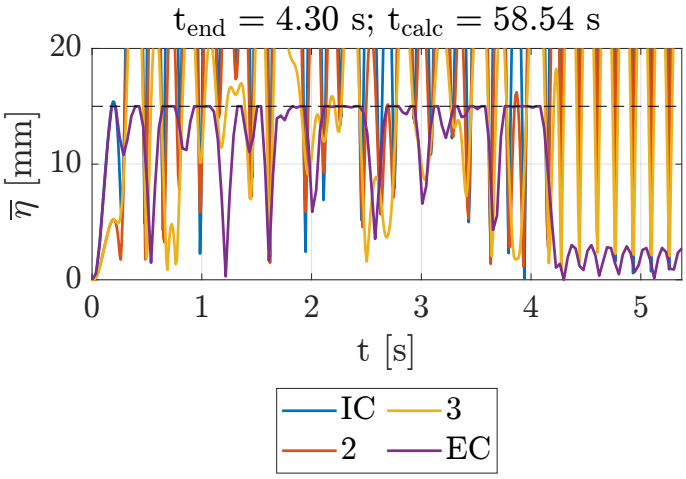


(c) $\overline{\eta}_{lim}$ on IC and EC.

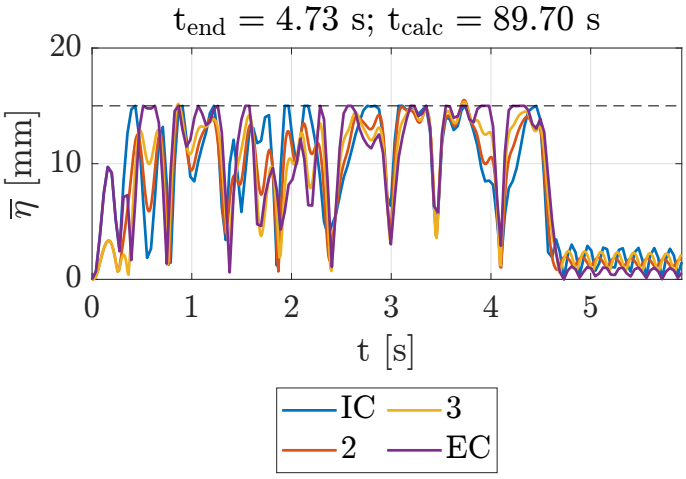
Figure A.4: Trajectory 1B, 6 containers.



(a) $\overline{\eta}_{lim}$ on all.

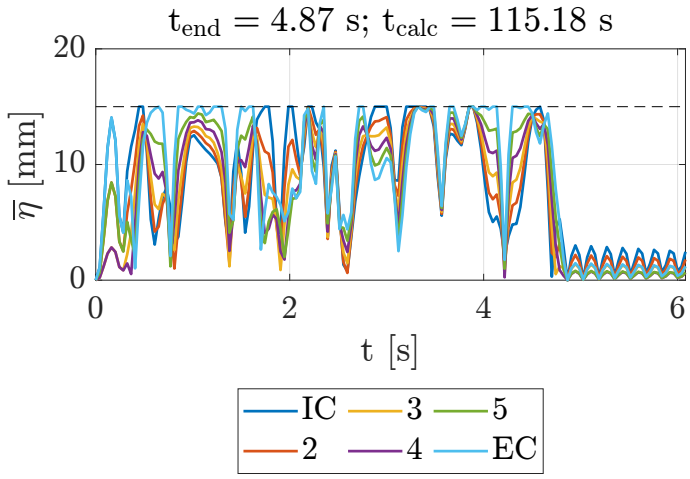


(b) $\overline{\eta}_{lim}$ on EC.

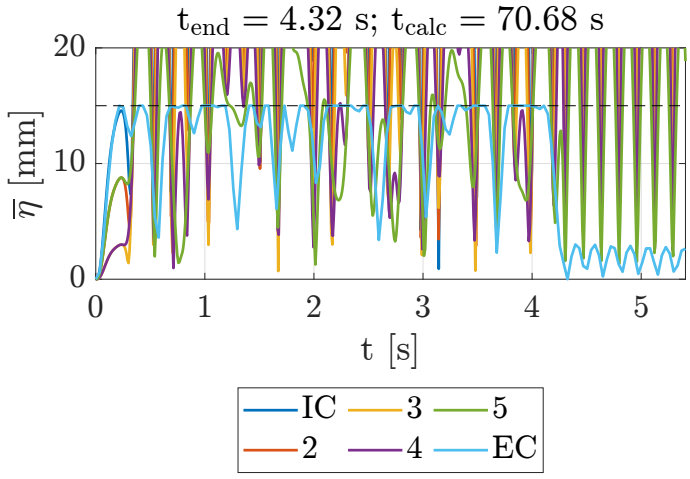


(c) $\overline{\eta}_{lim}$ on IC and EC.

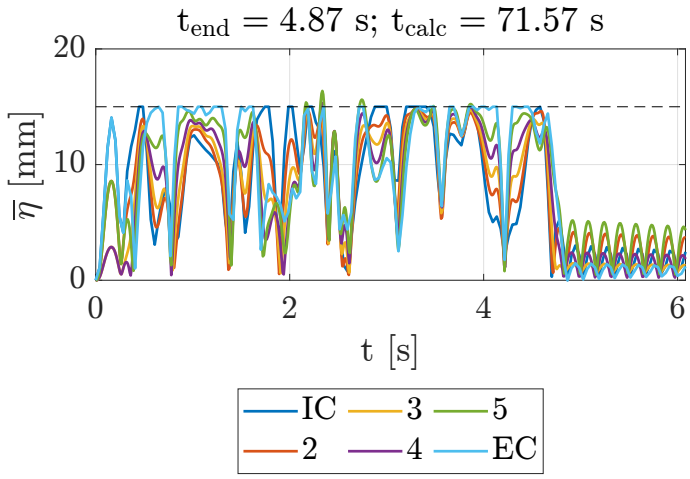
Figure A.5: Trajectory 2, 4 containers.



(a) $\overline{\eta}_{lim}$ on all.



(b) $\overline{\eta}_{lim}$ on EC.



(c) $\overline{\eta}_{lim}$ on IC and EC.

Figure A.6: Trajectory 2, 6 containers.

Appendix B

Robot motion analysis

B.1 Serial manipulator kinematic

Given the reference frame attached to the robot EE, $O_e x_e y_e z_e$, its position and orientation w.r.t. a fixed reference frame $O_b x_b y_b z_b$ can be determined using the transformation matrix:

$$ {}^b\mathbf{T}_e = \begin{bmatrix} {}^b\mathbf{R}_e & {}^b\mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^b\mathbf{n}_e & {}^b\mathbf{s}_e & {}^b\mathbf{a}_e & {}^b\mathbf{r} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.1} $$

where $\mathbf{n}_e$, $\mathbf{s}_e$, $\mathbf{e}_e$ are the unit vectors of the x_e, y_e, z_e axes, respectively, which form the columns of the rotation matrix $\mathbf{R}_e$, $\mathbf{r}$ is the position vector of O_e w.r.t. O_b, and finally ${}^b( )$ indicates the vector projection on reference frame b.
Since the position and orientation of the EE depend on the angles assumed by each individual joint, the matrix $\mathbf{T}_{be}$ can be written as a function of these angles. For this purpose, the *Denavit-Hartenberg* notation is used. Referring to figure B.1, four parameters can be identified:

- d_i: distance between the axes X_{i-1} and X_i, along the axis Z_{i-1};
- ϑ_i: the clockwise rotation angle between the axes X_{i-1} and X_i around the axis Z_{i-1};
- a_i: the distance between the axes Z_{i-1} and Z_i, along the axis X_{i-1};
- α_i: the clockwise rotation angle between the axes Z_{i-1} and Z_i around the axis X_{i-1}.

If the i-th joint is a revolute joint (as in our case), the joint parameter (q_i) is represented by ϑ_i, while it is represented by d_i if it is prismatic. The other parameters are constants that characterize the manipulator. The transformation matrix from the reference frame i to $i-1$ can be written as:

$$ {}^{i-1}\mathbf{T}_i(q_i) = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} c_{\alpha_i} & s_{\vartheta_i} s_{\alpha_i} & c_{\vartheta_i} a_i \\ s_{\vartheta_i} & s_{\vartheta_i} c_{\alpha_i} & -c_{\vartheta_i} s_{\alpha_i} & s_{\vartheta_i} a_i \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.2} $$

The overall transformation matrix of the manipulator can thus be written, from the reference frame 0 to the frame 6 of the last link (in our case, we have a 6-DOF manipulator,

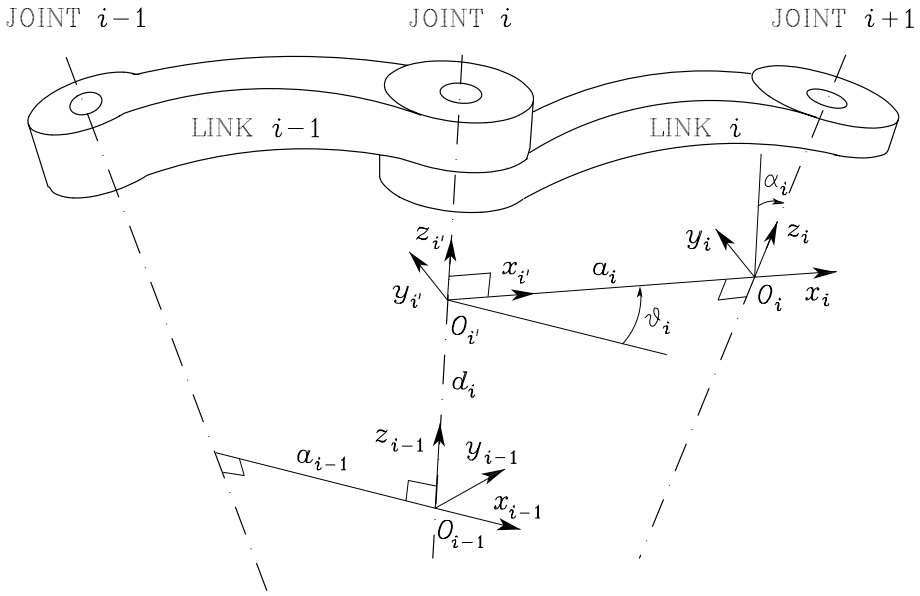


Figure B.1: Denavit Hartemberg parameters, from [45].

meaning 6 joints and moving links):

$$ {}^0\mathbf{T}_6(\mathbf{q}) = \prod_{i=1}^{6} {}^{i-1}\mathbf{T}_i(q_i), \tag{B.3} $$

where $\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_6 \end{bmatrix}^T \in \mathbb{R}^6$ is the vector containing the joint variables.
The matrix in equation (B.1) can thus be written as a function of the joint variables, as required:

$$ {}^b\mathbf{T}_e(\mathbf{q}) = {}^b\mathbf{T}_0 \cdot {}^0\mathbf{T}_6(\mathbf{q}) \cdot {}^6\mathbf{T}_e, \tag{B.4} $$

where ${}^b\mathbf{T}_0$ and ${}^6\mathbf{T}_e$ are constant homogeneous transformation matrices that describe, respectively, the pose of the frame 0 w.r.t. a fixed base frame b, and the pose of the frame e, attached to the EE, w.r.t. frame 6.
The procedure described thus far is known as *forward kinematics*, meaning that the pose of the robot EE is determined based on the joint variables. *Inverse kinematics* describes the opposite problem, i.e., starting from the transformation matrix, the joint variables must be determined. If the manipulator has a spherical wrist, as in our case, it is possible to decouple the position problem from the orientation problem, greatly simplifying the task, because the position is only determined by the first three joints, and the orientation is only determined by the last three.

B.2 Serial manipulator differential kinematic

In this section, the joint velocities of the robot will be determined, i.e., the vector $\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 & \dots & \dot{q}_6 \end{bmatrix}^T$, starting from the linear and angular velocities of the robot EE.
Referring to the translational and rotational part of equation (B.1), the following are defined:

- the linear velocity of the EE w.r.t. the reference frame b:

$$ {}^b\dot{\mathbf{r}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{d\,{}^b\mathbf{r}(\mathbf{q})}{dt} = \frac{d\,{}^b\mathbf{r}(\mathbf{q})}{d\mathbf{q}}\dot{\mathbf{q}}, \tag{B.5} $$

- the angular velocity of the EE w.r.t. the reference frame b:

$$ {}^{b}\boldsymbol{\omega} = \begin{bmatrix} \omega_{x_b} \\ \omega_{y_b} \\ \omega_{z_b} \end{bmatrix}, \tag{B.6} $$

where the three components of ${}^{b}\boldsymbol{\omega}$ are defined starting from ${}^{b}\mathbf{R}_{e}(\mathbf{q})$:

$$ {}^{b}\tilde{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_{z_b} & \omega_{y_b} \\ \omega_{z_b} & 0 & -\omega_{x_b} \\ -\omega_{y_b} & \omega_{x_b} & 0 \end{bmatrix} = {}^{b}\dot{\mathbf{R}}_{e}(\mathbf{q},\dot{\mathbf{q}})\,{}^{b}\mathbf{R}_{e}^{T}(\mathbf{q}), \tag{B.7} $$

hence:

$$ {}^{b}\boldsymbol{\omega} = {}^{b}\boldsymbol{\omega}(\mathbf{q},\dot{\mathbf{q}}). \tag{B.8} $$

Therefore, from the knowledge of $\mathbf{q}$ and $\dot{\mathbf{q}}$, it is possible to determine the velocities of the EE, and vice versa. The relationship between the following quantities is given by the *geometric Jacobian* (whose definition is provided in [45]):

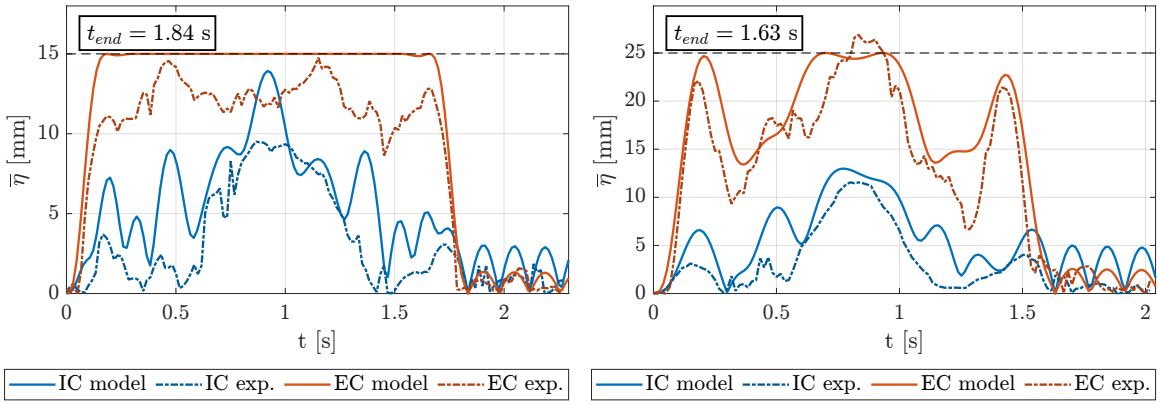
$$ {}^{b}\mathbf{v} = \begin{bmatrix} {}^{b}\dot{\mathbf{r}} \\ {}^{b}\boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \tag{B.9} $$

from which:

$$ \dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\,{}^{b}\mathbf{v}. \tag{B.10} $$

Appendix C

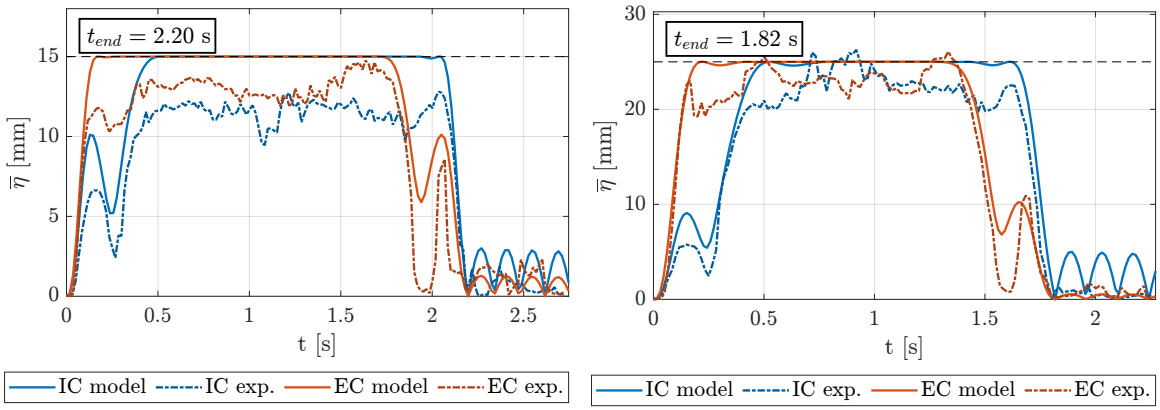
Experimental results for Trajectory Volumes 1A and 1B



(a) Optimized motion law with $\overline{\eta}_{lim}$ = 15 mm.

(b) Optimized motion law with $\overline{\eta}_{lim}$ = 25 mm.

Figure C.1: Trajectory Volumes 1A.



(a) Optimized motion law with $\overline{\eta}_{lim}$ = 15 mm.

(b) Optimized motion law with $\overline{\eta}_{lim}$ = 25 mm.

Figure C.2: Trajectory Volumes 1B.

Considering the trajectories referred to, in accordance with what was stated in chapter 4, as Trajectory Volumes 1A and Trajectory Volumes 1B, the results obtained by imposing a sloshing height limit of both $\overline{\eta}_{lim}$ = 15 mm and $\overline{\eta}_{lim}$ = 25 mm are reported below. In each subplot of figure C.1 (for Trajectory Volumes 1A) and figure C.2 (for Trajectory Volumes 1B), the sloshing heights of the IC (blue line) and EC (red line) containers are shown. The continuous lines represent the trends predicted by the NL model, while the dashed lines correspond to the experimental measurements. It can be observed that the trends predicted by the model and those obtained experimentally are very similar, indicating that the NL model performs well for these applications. Additionally, it is noted that the sloshing height limits are respected, within a margin of a few millimeters, for each trajectory.

Appendix D

Generative AI tools usage statement

During the preparation of this work, the generative artificial intelligence tool ChatGPT-4 (version released in March 2024) was used. This tool was primarily employed between September 2024 and March 2025, with the aim of enhancing the clarity and grammatical accuracy of certain paragraphs and assisting in the creation of graphs, specifically in chapters 3 and 4. However, it did not replace the critical, analytical, and creative contribution of the author to the development of this thesis.

Acknowledgments

The author would like to thank the Laboratory of Automation and Robotics (LAR) at the University of Bologna, and in particular Simone Soprani, for providing the robot used for the experiments.
A sincere thanks also to Roberto Di Leva and professor Marco Carricato for their guidance and support in the research process.

List of Figures

Bibliography

[1] S. B. Coogan and S. Green, "Critical Review of Damping Prediction Methods for Annular Ring Slosh Baffles," in *Proceedings of the 2019 AIAA Propulsion and Energy 2019 Forum and Exposition, Indianapolis, IN, USA*, pp. 1–16, 2019.

[2] Z. Ge, Y. Li, and S. Ma, "Attitude Stabilization of Rocket Elastic Vibration Based on Robust Observer," *Aerospace*, vol. 9(12), no. 765, pp. 1–20, 2022.

[3] T. Aierken, S. Li, and M. Tingqian, "Multi-Objective Optimization Study of LNG Tank Truck Baffles under the Most Hazardous Transport Condition based on a Combined Approximate Model," *Ocean Engineering*, vol. 314, no. 119713, pp. 1–20, 2024.

[4] M. M. S. Lee, C. Chung, and S.-G. Lee, "Sloshing Reduction Using Floating Mat for Floating LNG Cargo Containment System," in *Proceedings of the 2016 Offshore Technology Conference Asia*, pp. 4035–4043, 2016.

[5] G. R. Patil and K. D. Singh, "Evaluation of Sloped Bottom Tuned Liquid Damper for Reduction of Seismic Response of Tall Buildings," *Journal of The Institution of Engineers (India): Series A*, vol. 97, no. 4, pp. 385–394, 2016.

[6] S. S. Roy and K. C. Biswal, "Multiple Sloped Wall Tuned Liquid Dampers for Vibration Control of Structure Excited by Near and Far-Fault Earthquakes," *Journal of Vibration Engineering amp; Technologies*, vol. 12, no. 7, pp. 8465–8484, 2024.

[7] G. S. Brar and S. Singh, "An Experimental and CFD Analysis of Sloshing in a Tanker," *Procedia Technology*, vol. 14, pp. 490–496, 2014.

[8] E. Frosina, A. Senatore, A. Andreozzi, G. Marinaro, D. Buono, G. Bianco, D. Auriemma, F. Fortunato, F. Damiano, and P. Giliberti, "Study of the Sloshing in a Fuel Tank Using CFD and EFD Approaches," in *Proceedings of the 2017 ASME/BATH 2017 Symposium on Fluid Power and Motion Control*, pp. 1–13, 2017.

[9] X. Cao, F. Ming, and A. Zhang, "Sloshing in a Rectangular Tank Based on SPH Simulation," *Applied Ocean Research*, vol. 47, pp. 241–254, 2014.

[10] J. Shao, H. Li, G. Liu, and M. Liu, "An Improved SPH Method for Modeling Liquid Sloshing Dynamics," *Computers amp; Structures*, vol. 100–101, pp. 18–26, 2012.

[11] R. A. Ibrahim, *Liquid Sloshing Dynamics: Theory and Applications*. Cambridge University Press, 2005.

[12] W. Aribowo, T. Yamashita, and K. Terashima, "Integrated Trajectory Planning and Sloshing Suppression for Three-Dimensional Motion of Liquid Container Transfer Robot Arm," *Journal of Robotics*, vol. 2015, pp. 1–15, 2015.

[13] J. Reinhold, M. Amersdorfer, and T. Meurer, "A Dynamic Optimization Approach for Sloshing Free Transport of Liquid Filled Containers using an Industrial Robot," in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China*, pp. 2336–2341, IEEE, 2019.

[14] L. Moriello, L. Biagiotti, C. Melchiorri, and A. Paoli, "Control of Liquid Handling Robotic Systems: A Feed-Forward Approach to Suppress Sloshing," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore*, pp. 4286–4291, 2017.

[15] L. Moriello, L. Biagiotti, C. Melchiorri, and A. Paoli, "Manipulating Liquids with Robots: A Sloshing-Free Solution," *Control Engineering Practice*, vol. 78, pp. 129–141, 2018.

[16] L. Biagiotti, D. Chiaravalli, L. Moriello, and C. Melchiorri, "A Plug-In Feed-Forward Control for Sloshing Suppression in Robotic Teleoperation Tasks," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain*, pp. 5855–5860, 2018.

[17] H. Bauer, "Nonlinear Mechanical Model for the Description of Propellant Sloshing," *AIAA Journal*, vol. 4, pp. 1662–1668, 1966.

[18] L. Guagliumi, A. Berti, E. Monti, and M. Carricato, "A Simple Model-Based Method for Sloshing Estimation in Liquid Transfer in Automatic Machines," *IEEE Access*, vol. 9, pp. 129347–129357, 2021.

[19] L. Guagliumi, A. Berti, E. Monti, and M. Carricato, "Anti-Sloshing Trajectories for High-Acceleration Motions in Automatic Machines," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 7, pp. 1–14, 2022.

[20] R. Di Leva, M. Carricato, H. Gattringer, and A. Müller, "Sloshing Dynamics Estimation for Liquid-filled Containers under 2-Dimensional Excitation," in *Proceedings of the 10th ECCOMAS Thematic Conference on MULTIBODY DYNAMICS, Budapest, Hungary*, pp. 80–89, 2021.

[21] R. Di Leva, M. Carricato, H. Gattringer, and A. Müller, "Sloshing Dynamics Estimation for Liquid-Filled Containers Performing 3-Dimensional Motions: Modeling and Experimental Validation," *Multibody System Dynamics*, vol. 56, no. 2, pp. 153–171, 2022.

[22] R. Di Leva, S. Soprani, F. Vittuari, D. Ceccarelli, G. Palli, L. Biagiotti, and M. Carricato, "Sloshing-Height Estimation for Liquid-filled Containers under Schoenflies Motions: Modeling and Experimental Validation," *in preparation*, 2025.

[23] R. Di Leva, M. Carricato, H. Gattringer, and A. Muller, "Time-Optimal Trajectory Planning for Anti-Sloshing 2-Dimensional Motions of an Industrial Robot," in *Proceedings of the 20th International Conference on Advanced Robotics (ICAR), Ljubljana, Slovenia*, pp. 32–37, 2021.

[24] L. Guagliumi, A. Berti, E. Monti, and M. Carricato, "A Software Application for Fast Liquid-Sloshing Simulation," in *Proceedings of the 4th International Conference of IFToMM Italy, Naples, Italy*, pp. 819–828, 2022.

[25] L. Biagiotti and C. Melchiorri, "FIR Filters for Online Trajectory Planning with Time - and Frequency - Domain Specifications," *Control Engineering Practice*, vol. 20(12), pp. 1385–1399, Dec. 2012.

[26] L. Biagiotti, C. Melchiorri, and L. Moriello, "Optimal Trajectories for Vibration Reduction Based on Exponential Filters," *IEEE Transactions on Control Systems Technology*, vol. 24(2), pp. 609–622, 2015.

[27] N. C. Singer and W. P. Seering, "Preshaping Command Inputs to Reduce System Vibration," *ASME Journal of Dynamics Systems, Measurements, and Control*, vol. 112(1), pp. 76–82, 1990.

[28] W. Aribowo, T. Yamashita, K. Terashima, and H. Kitagawa, "Input Shaping Control to Suppress Sloshing on Liquid Container Transfer using Multi-Joint Robot Arm," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan*, pp. 3489–3494, 2010.

[29] J. Lawrence and W. Singhose, "Command Shaping Slewing Motions for Tower Cranes," *Journal of Vibration and Acoustics*, vol. 132, no. 1, pp. 1–11, 2010.

[30] L. Biagiotti, C. Melchiorri, and L. Moriello, "Damped Harmonic Smoother for Trajectory Planning and Vibration Suppression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 2, pp. 626–634, 2020.

[31] E. Khorshid and A. Al-Fadhli, "Optimal Command Shaping Design for a Liquid Slosh Suppression in Overhead Crane Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 143, no. 2, pp. 1–13, 2020.

[32] F. Brasina, L. Guagliumi, R. Di Leva, and M. Carricato, "Anti-sloshing Motion Laws for One-Dimensional Piecewise Trajectories," in *Proceedings of the 9th European Conference on Mechanism Science, Padua, Italy*, pp. 153–162, 2024.

[33] K. Zauner, H. Gattringer, and A. Müller, "Time-Optimal Transport of Loosely Placed Liquid Filled Cups Along Prescribed Paths," in *Proceedings of the 33rd International Conference on Robotics in Alpe-Adria-Danube Region, Cluj-Napoca, Romania*, pp. 411–419, 2024.

[34] H. Gattringer, A. Müller, S. Weitzhofer, and M. Schörgenhumer, "Point to point time optimal handling of unmounted rigid objects and liquid-filled containers," *Mechanism and Machine Theory*, vol. 184, no. 105286, pp. 1–11, 2023.

[35] J. Andersson, J. Gillis, G. Horn, J. Rawlings, and M. Diehl, "CasADi: a Software Framework for Nonlinear Optimization and Optimal Control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[36] Bauer, H.F., "Tables of zeros of cross product bessel functions," *Mathematics of Computation*, vol. 18, no. 85, pp. 128–135, 1964.

[37] S. Truxillo, *Delay Times in Spin-Up From Rest of a Viscous Fluid in a Cilindrical Container*. PhD thesis, Louisiana State University Libraries, 1969.

[38] Çengel Yunus A. and J. M. Cimbala, *Fluid mechanics*. New York, NY: McGraw-Hill Education, 2017.

[39] H. Baruh, *Applied dynamics*. CRC Press, 2019.

[40] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer Berlin, 2010.

[41] L. Piegl and W. Tiller, *The NURBS Book*. Springer Berlin, 1997.

[42] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer New York, 2006.

[43] R. Di Leva, *Trajectory Planning of Single and Dual-Arm Robots for Time-Optimal Handling of Liquids and Objects*. PhD thesis, Alma Mater Studiorum University of Bologna, 2023.

[44] T. Krityakierne and C. A. Shoemaker, "SOMS: SurrOgate MultiStart Algorithm For use with NonLinear Programming for Global Optimization," *International Transactions in Operational Research*, vol. 24, no. 5, pp. 1139–1172, 2015.

[45] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer London, 2009.