POD-based State-Constrained Economic Model Predictive Control of Convection-Diffusion Phenomena

Doctoral thesis
for the award of the academic degree
Doctor of Natural Sciences
(Dr.rer.nat.)

at the

Universität Konstanz

Faculty of Sciences
Department of Mathematics and Statistics

submitted by
Luca Mechelli

Day of the oral examination: 8th November 2019

Referee: Prof. Dr. Stefan Volkwein, University of Konstanz
Referee: Prof. Dr. Lars Grüne, University of Bayreuth
Abstract

Motivated by an application to energy efficient building, in this thesis optimal control of a linear time-dependent convection-diffusion equation is considered together with bilateral control and state constraints. To obtain regular Lagrange multipliers, two different approaches are investigated. The first one is called virtual control approach and it consists in introducing an artificial control, which allows replacing the pointwise state constraints with mixed constraints. The regularized problem is then solved with the Primal Dual Active Set Strategy (PDASS). The second applied technique is the augmented Lagrangian method: penalization terms dependent on the state constraints are added to the cost functional and many only control constrained problems are iteratively solved applying projected gradient or projected BFGS methods. To speed-up the computational time proper orthogonal decomposition (POD) is considered together with an a-posteriori error analysis, which ensures the quality of the POD approximation. Furthermore, model predictive control (MPC) helps to handle the long-time horizon and the problem parameter changes in an efficient way. At last, numerical simulations show the advantages and disadvantages of these techniques.
Zusammenfassung

Acknowledgements

I gratefully acknowledge financial support over the last three years by the Deutsche Forschungsgesellschaft (DFG) grant VO 1658/4-1 Reduced-Order Methods for Nonlinear Model Predictive Control. Moreover, I would like to thank everyone who has been close to me during my doctoral studies. Without their moral support, I could not have written this thesis.

At first, I want to thank my supervisor Prof. Dr. Stefan Volkwein. His precious advice and his willingness to help always made me feel welcomed. With due care, he has always been able to advise and correct me during this journey. I owe him my professional growth and the awareness of my future plans. I express gratitude also to my colleagues for the light and serene atmosphere, even in the most tiring working days. I thank them for the fruitful discussions, the evenings spent together and the huge help with the German language and bureaucracy. In particular, I wish to thank my officemates, that have stood my extreme talkativeness and daily anxiety. Thank you Dennis, Freya, Gabriele, Georg, Giulia, Jianjie, Sabrina and Stefan. I would like to thank also Prof. Dr. Lars Grüne, Prof. Dr. Michael Hinze, Prof. Dr.-Ing. Thomas Meurer, Prof. Dr.-Ing. Gianluigi Rozza and their respective working groups. I have found a convivial environment, I could enrich my scientific knowledge and undertake successful collaborations in each of these groups. I thank especially my two DFG project partners Simon Pirkelmann and Julian Andrej for the time spent together. Despite coming from different backgrounds, we have been able to unify our skills at the service of one another in a friendly surrounding.

In the following lines I would like to thank my family speaking to them in italian. Grazie mamma e papà, grazie nonna. So quanto questi anni siano stati difficili per voi, ma, nonostante la distanza, vi ho sempre sentiti vicini ed ho potuto contare sul vostro amore. La volontà di rendervi fieri mi ha sempre spinto ad impegnarmi al massimo. I want to thank also my lifelong friends, cornerstones of my life. Everytime I have been back home, they have done everything they could to spend time together
with me. Thank you, Cristiano, Davide, Francesco, Michela, Simone and Valentina. Thanks also to my travel companions in this adventure in Germany, Nicola and Sara, that I met accidentally and luckily three years ago at our first German course. Although we have not reached a high German level, we have surely created a strong bond of friendship, which I will always cherish. I wish to thank my tandem partner Svenja as well. The time spent with her was fun, trying to teach each other how to correctly speak our respective languages, avoiding the interferences of our mutual dialects. Moreover, I would like to thank my girlfriend’s family to let her visit me whenever is possible and to make me feel at home even when I stay at their place. Thank you, Chiara, Cristina and Graziano.

The most important person to thank is my girlfriend Martina, to whom this work is dedicated. Grazie amore mio per questi anni. I could not have reached this point without your constant presence and support. Every text message, every video call and every visit have given me a powerful energy charge. Sometimes the distance has a big impact, but your love is always present. Grazie.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Outline</td>
<td>4</td>
</tr>
<tr>
<td>1 The linear-quadratic optimal control problem</td>
<td>7</td>
</tr>
<tr>
<td>1.1 The state equation</td>
<td>7</td>
</tr>
<tr>
<td>1.1.1 Maximal Regularity</td>
<td>13</td>
</tr>
<tr>
<td>1.2 The state-constrained optimal control problem</td>
<td>16</td>
</tr>
<tr>
<td>1.2.1 First-order optimality conditions</td>
<td>18</td>
</tr>
<tr>
<td>1.2.2 Gradient of the reduced cost functional</td>
<td>20</td>
</tr>
<tr>
<td>1.3 The virtual control approach</td>
<td>22</td>
</tr>
<tr>
<td>1.3.1 First-order optimality conditions</td>
<td>25</td>
</tr>
<tr>
<td>1.3.2 Gradient of the reduced cost functional</td>
<td>26</td>
</tr>
<tr>
<td>1.3.3 Regularization Error</td>
<td>28</td>
</tr>
<tr>
<td>1.3.4 Primal-dual active set strategy</td>
<td>43</td>
</tr>
<tr>
<td>1.4 The augmented Lagrangian method</td>
<td>52</td>
</tr>
<tr>
<td>1.4.1 Gradient of the reduced augmented Lagrangian</td>
<td>55</td>
</tr>
<tr>
<td>1.4.2 Proof of convergence</td>
<td>59</td>
</tr>
<tr>
<td>1.5 Numerical tests</td>
<td>66</td>
</tr>
<tr>
<td>1.5.1 Tracking term</td>
<td>69</td>
</tr>
<tr>
<td>1.5.2 Economic optimal control</td>
<td>73</td>
</tr>
<tr>
<td>1.5.3 PDASS matrix-free implementation</td>
<td>76</td>
</tr>
</tbody>
</table>

2 Proper Orthogonal Decomposition for linear-quadratic OCP | 79 |
| 2.1 Basics | 79 |
| 2.2 POD-based primal-dual active set strategy | 82 |
| 2.3 POD-based augmented Lagrangian method | 85 |
| 2.4 A-posteriori error estimates | 88 |
3 POD-based Model Predictive Control for linear-quadratic OCP 101
  3.1 Model Predictive Control ................................. 101
  3.2 MPC-POD algorithms ...................................... 104
  3.3 Numerical tests ............................................ 108
    3.3.1 Feedback versus open-loop control .................. 109
    3.3.2 MPC and POD ........................................... 111
Introduction

Nowadays the study and design of energy efficient buildings is an important part of the scientific research. In [109], for example, they investigate how the structure of a building influences its efficiency. In particular, they review all the possible parameters which has to be considered in the design process. Instead, in [34] they study the possibility to control the heating process with real-time measurement of the occupancy. This approach is convenient to treat each room differently, basing the optimal control strategy on the numbers of people in each area. Furthermore, in [108, 124] a model predictive control strategy is used to optimize the temperature inside the building taking in account the weather forecast, which may change as soon as the time passes. The previously cited works utilize ordinary differential equations (ODEs) to model the time evolution of the temperature profile. Although this is a valid approach, model based on partial differential equations (PDEs), e.g., parabolic equations, allows a deeper understanding of the physical phenomena involved. In particular, the diffusion-convection equation is suitable for room heating, since it describes, e.g., how hot regions progressively cool down in presence of colder regions and vice versa (diffusion), how density differences cause hot air to rise and cold air to descend (natural convection) and how heating devices, e.g., convection heaters, force the air to move (artificial convection). Considering energy efficient heating, the goal is not only to model how the temperature profile evolves, but also to control such temperature through physical devices, so that it remains inside a comfortable range. Therefore, an optimal control approach has to be considered to treat the problem; cf., e.g., [110].

The optimal control of parabolic equations is an established field today. Many mathematicians have contributed to this field over decades. A milestone is the book of J.L. Lions [92], while more recently [67, 126] resume the standard theoretical approaches and numerical techniques. In this work we are particularly interested in pointwise state constrained problems, i.e., as already said, we want to keep
the solution of the convection-diffusion equation inside predefined bounds. These problems are particularly challenging, because of the low regularity of the solution to the adjoint equation and of the Lagrange multipliers. In fact, these multipliers belong to the space of Radon measures [23, 27, 30, 32, 39, 115, 117] instead of the set of Lebesgue square integrable functions $L^2$ [67, 126], affecting the regularity of the dual variable. Moreover, if one is interested in semilinear parabolic equations and in deriving second-order necessary and sufficient optimality conditions, the mathematical challenges grow; cf. [28, 82, 116].

For the previous reasons, regularization techniques or penalty methods can be applied to approximate the solution of the pointwise state constrained problem, gaining regular ($L^2$) Lagrange multipliers. Let us mention at first the work of I. Neitzel, A. Rösch and F. Tröltzsch [102, 103, 121, 125], where the authors study two regularization approaches: the Lavrentiev regularization and the Moreau-Yosida one. The first one consists in replacing the pointwise state constraints with mixed state-control constraints, i.e., the control-dependent solution of the parabolic PDE plus the control has to be inside the bounds. This optimization problem can be solved with the Primal-Dual Active Set Strategy (PDASS), which is equivalent to a semi-smooth Newton method; cf. [22, 63, 64, 74, 128]. Following the same idea, one can also consider the virtual control (VC) approach introduced for an elliptic optimal control problem by K. Krumbiegel and A. Rösch in [83]. The difference is in introducing an artificial control variable (different from the controls of the problem) in the mixed constraints. For example, this is useful in the case of boundary control, where the control and the state variables do not belong to the same space, therefore the Lavrentiev regularization can not be performed. Instead, the Moreau-Yosida approach adds a quadratic penalization term depending on the state constraints to the cost functional. Therefore, the solution of the original problem can be computed by solving iteratively several problems that are only control constrained. This last approach is equivalent to the augmented Lagrangian method (ALM); cf. [20, 21, 72]. Recently, V. Karl, I. Neitzel and D. Wachsmuth proposed a variant of the ALM for linear and semilinear elliptic PDEs [79, 80], basing the choice to update or not the multipliers on predefined residual quantities.

Numerically, the solution of the regularized problem can be computed with the Finite Element method (FE) in space and with a numerical method for ODEs in time, e.g., explicit or implicit Euler. For a large number of FE nodes and of time steps, solving state and adjoint equations is costly in terms of computational time and memory, in particular with the PDASS. Therefore, one can think on a
model order reduction technique. There are many possibility to perform model order reduction for a linear PDE, such as balanced truncation, proper orthogonal decomposition (POD) and the reduced basis methods. In the balanced truncation method the reduced-order space is built removing the states which are difficult to reach or observe; cf., e.g., [19, 58, 97, 100]. Instead, the reduced basis method is designed for parametric PDEs and it consists in two separated parts called offline and online phases. In the first step, the most relevant parameters are selected through a Greedy algorithm based on an a-posteriori error estimator, in order to approximate the parametric solution manifold. In the online phase, the solution can then be evaluated fast for different parameters. To this field belongs the following contributions [16, 40, 59, 105, 111], to name just a few. The POD method is based on previously computed information on the system dynamics, i.e., the so-called snapshots. The idea is to compute the first $\ell$ basis of the snapshots subspace such that the distance between the snapshots and their projection on the reduced space is minimized. This minimization problem can be solved equivalently computing the eigenvalues of a corresponding self-adjoint compact non-negative operator. The literature about the POD method has grown exponentially in the last twenty years, among others we refer to [44, 46, 57, 68, 69, 85, 86]. Furthermore, to certify the quality of the approximation of the reduced-order model, it is necessary an a-posteriori error estimator, that allows estimating the error without computing the solution of the full-order model; cf., e.g., [13, 56, 55, 113, 127]. The a-posteriori error estimates are also useful to select the most relevant parameters in the reduced basis method [31, 89, 120, 129, 130] and to develop a basis update strategy for the POD algorithm [2, 14, 45]. Let us mention that another possibility to update the POD basis is to consider Optimality System POD (OS-POD) and Trust Region POD (TR-POD); see [43, 87] and [10, 122], respectively.

In optimal control it is also important to react to parameters changes and to correct model uncertainties. To achieve this goal, one has to consider closed-loop controls, also known as feedback controls. These controls depend on the state variable and are robust to external perturbation of the system. If this feedback relationship is not established, we have an open-loop control, which then depends only on the time variable. Starting from the work of Bellman [17], the theory of closed-loop optimal control is one of the most active research field of the last century and it is still object of interest nowadays. The two most common techniques are the Hamilton-Jacobi-Bellman approach (HJB) and Model Predictive Control (MPC), which is also known as receding horizon control. The HJB approach is based on
the dynamic programming principle [17] applied to the value function, which is the
viscosity solution to a non-linear HJB equation; cf. [15]. The drawback of this
approach are the numerical challenges, e.g., the curse of dimensionality, that makes
costly (or even impossible) the computation of the value function. To overcome
this problem, different numerical strategies were developed during the past years,
see, e.g., [1, 24, 25, 26, 37, 47, 84]. As well as the HJB approach, MPC relies on
the dynamic programming principle, but it optimizes and stabilizes a given system
around a predefined equilibrium through prediction of its future behaviour. The
feedback controls are computed iterating the following procedure: at first we solve
the optimal control problem for a fixed prediction horizon length, we store the
open-loop optimal control at the first time step and then we shift the horizon for
the next open-loop problem, updating if necessary the parameters of the system; cf.
[52, 73, 81, 94, 114]. In recent time, starting from the work of D. Angeli, R. Amrit
and J.B. Rawlings [5, 7, 8], a new branch of MPC has been developed, the so-called
economic MPC. In economic MPC we do not want to track the state variable to
a predefined equilibrium, but we are interested in other type of quantities, such
as energy consumption, power production, yield of a substance and so on. Strict
dissipativity and, thus, the turnpike property are sufficient to ensure asymptotic
stability for the closed-loop trajectory of the economic MPC; cf. [48, 50, 53, 54, 101].
Let us point out that economic MPC is particularly useful in our application, since
we do not know a-priorily which state is the best for our purpose.

The idea to combine feedback controls and model order reduction has already
been investigated nowadays. Regarding HJB feedback control combined with a
model order reduction approach we mention [88, 91]. For POD-based MPC see, e.g.,
[38, 70, 71]. At last, in [36] the authors consider a reference trajectory computed
by MPC to shrink the computational domain of the HJB equation, speeding up the
computational time with POD.

**Outline**

This work is focused on the regularization of the pointwise state constraints for
an optimal boundary control problem subjected to the convection-diffusion equation
and on how efficiently combine these techniques with POD and MPC. Moreover, the
discussion is oriented on an energy efficient building application: the convection-
diffusion equation models the temperature inside a room, which can be controlled
by boundary heating devices and it is subjected to the influence of the outside
temperature and artificially generated airflow.

In Chapter 1 we treat the regularization of the pointwise state constraints, focusing on the VC approach and on the ALM.

- In Section 1.1, we introduce the state equation and we discuss under which conditions the solution is continuous. This maximal regularity result is an extension of the work presented in [126], where the proof is given for a diffusion parabolic differential operator and Neumann boundary conditions. In this thesis, we consider a convection-diffusion parabolic operator and Robin boundary conditions. Let us mention that a more general result for semilinear parabolic convection-diffusion-reaction equations, but with different proof, is shown in [27].

- In Section 1.2, we introduce the pointwise state-constrained optimal control problem and we briefly illustrate the well-known results of low regularity for the adjoint equation; see, e.g., [27, 67].

- Section 1.3 is entirely dedicated to the VC approach. At first, we introduce this regularization technique and we derive first-order sufficient and necessary optimality conditions for the regularized problem. Afterwards, we prove the convergence of the regularized solution to the one of the problem with pointwise state constraints, extending the proof in [83]. Let us mention that the proof presented in [83] is only for an optimal control problem with an elliptic diffusion-reaction equation, Neumann boundary control and unilateral state constraint. Instead, in our case we have a parabolic convection-diffusion operator, Robin boundary controls and bilateral state constraints. In the last part of the section, we show how to apply the PDASS to this problem and face numerical challenges, according to the work of M. Hintermüller, K. Ito and K. Kunisch [63, 74, 75]. This section refers on our preliminary work done in [95].

- In Section 1.4, continuing the work done in [78], we apply the ALM as a second way to regularize the state-constrained problem. In addition to [78], we adapt the algorithm in [79, 80] to our different setting. In fact, in [80] the optimal control problem is subjected only to an elliptic diffusion-reaction equation with distributed control, homogeneous Neumann boundary conditions and unilateral state constraint. In [79], the authors study the same optimal control problem subjected to a semilinear elliptic PDE. Moreover, following the proof contained in [80] we show convergence for the adapted algorithm.
• Numerical simulations are reported in Section 1.5. This section is split in three parts: in the first part tracking-type cost functional is considered, while in the second subsection we choose an economic cost and in the last part we exploit the matrix-free structure of the PDASS algorithm, showing improvement on the computational cost in terms of memory consumption.

In Chapter 2, we apply the POD method to our linear-quadratic optimal control problem to gain computational time speed-up and decrease the memory consumption.

• In Section 2.1, we briefly report the basic theory of the POD method, in order to fix the notation.

• In Section 2.2 and in Section 2.3, we show how the POD method can be applied to the VC approach and to the ALM, respectively.

• In Section 2.4, we derive the a-posteriori error estimate for the POD-based VC approach and for the POD-based ALM. The first a-posteriori error estimate can be derived similarly to the one contained in [43]. Let us mention that compared to [43] we have additional constraints on the virtual control, that have to be taken into account in the proof. Instead, the a-posteriori error estimator for the POD-based ALM follows from standard arguments; cf., e.g., [127].

• Section 2.5 contains numerical tests, which show the effectiveness of the POD method and of the a-posteriori error estimators. Moreover, in addition to [78], we introduce a new way to generate the POD snapshots for the augmented Lagrangian algorithm.

In Chapter 3, we introduce MPC and we show how to combine it with POD and the a-posteriori error estimator, extending our work in [96]. As previously mentioned, this issue is already investigated in [36, 38], but the authors do not consider pointwise state constraints and economic MPC. In the last part of the chapter, numerical simulations illustrate the efficiency of the proposed strategies in terms of feedback reaction of the controls, reduction of the computational time and small violations of the state constraints. At the end of the work, we draw conclusions and comment on possible future work we are interested in.
Chapter 1

The linear-quadratic optimal control problem

In this chapter, we introduce the object of interest of this thesis: the linear-quadratic optimal control problem with bilateral control and state constraints. At first, we present the state equation and we explain under which assumptions its solution is continuous in time and space. Subsequently, we introduce the pointwise state constraints and comment about the regularity issues concerning the adjoint equation. Due to this reason, we show two regularization techniques, i.e., the virtual control approach and the augmented Lagrangian method, proving that their solution convergences to the original problem’s one as soon as their respective regularization parameters go to zero or to infinity. In the last part of this chapter, numerical simulations confirm the results predicted by the theory. In particular, we study tracking term type and economic cost functionals.

1.1 The state equation

Let $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$, be a bounded domain with Lipschitz-continuous boundary $\Gamma = \partial \Omega$. We suppose that $\Gamma$ is split into two disjoint subsets $\Gamma_c$ and $\Gamma_o$, where at least $\Gamma_c$ has nonzero (Lebesgue) measure. Further, let $H = L^2(\Omega)$ and $V = H^1(\Omega)$ endowed with their usual inner products

$$
\langle \varphi, \psi \rangle_H = \int_{\Omega} \varphi \psi \, d\mathbf{x}, \quad \langle \varphi, \psi \rangle_V = \int_{\Omega} \varphi \psi + \nabla \varphi \cdot \nabla \psi \, d\mathbf{x}
$$
1. The linear-quadratic optimal control problem

and their induced norms, respectively. For bounded and fixed $T > 0$ we set $Q = (0, T) \times \Omega$, $\Sigma_c = (0, T) \times \Gamma_c$ and $\Sigma_o = (0, T) \times \Gamma_o$. By $L^2(0, T; V)$ we denote the space of measurable functions from $[0, T]$ to $V$, which are square integrable, i.e.,

$$
\int_0^T \| \varphi(t) \|^2_V \, dt < \infty.
$$

When $t$ is fixed, the expression $\varphi(t)$ stands for the function $\varphi(t, \cdot)$ considered as a function in $\Omega$ only. The space $W(0, T)$ is defined as

$$
W(0, T) = \{ \varphi \in L^2(0, T; V) \mid \varphi_t \in L^2(0, T; V') \},
$$

where $V'$ denotes the dual of $V$. The space $W(0, T)$ is a Hilbert space supplied with the common inner product; c.f. [29, pp. 472-479]. For $m \in \mathbb{N}$ let $b_i : \Gamma_c \to \mathbb{R}$, $1 \leq i \leq m$, denote given control shape functions. For $U = L^2(0, T; \mathbb{R}^m)$ the set of admissible controls $u = (u_i)_{1 \leq i \leq m} \in U$ is given as

$$
U_{ad} = \{ u \in U \mid u_{ai}(t) \leq u_i(t) \leq u_{bi}(t) \text{ for } i = 1, \ldots, m \text{ and a.e. in } [0, T] \},
$$

where $u_a = (u_{ai})_{1 \leq i \leq m}$, $u_b = (u_{bi})_{1 \leq i \leq m} \in L^\infty(0, T; \mathbb{R}^m)$ are lower and upper bounds, respectively. Further, ‘a.e.’ stands for ‘almost everywhere’. Throughout the thesis we identify the dual $U'$ with $U$. This particular choice of $U_{ad}$ is justified by our application: since the controls may represent the heaters in the room or other devices, they are subjected to physical restrictions, such as limited heating power or minimum cooling temperature.

For any control $u \in U_{ad}$ the temperature (or state) $y$ is governed by the following state equation

$$
y_t(t, x) - \lambda \Delta y(t, x) + v(t, x) \cdot \nabla y(t, x) = 0, \quad \text{a.e. in } Q, \quad \lambda \frac{\partial y}{\partial n}(t, s) + \gamma_c y(t, s) = \gamma_c \sum_{i=1}^m u_i(t) b_i(s), \quad \text{a.e. on } \Sigma_c, \quad \lambda \frac{\partial y}{\partial n}(t, s) + \gamma_o y(t, s) = \gamma_o y_{out}(t), \quad \text{a.e. on } \Sigma_o, \quad y(0, x) = y_o(x), \quad \text{a.e. in } \Omega. \tag{1.1}
$$

We suppose the following hypotheses for the data in (1.1):

**Assumption 1.1.** We assume that $\lambda > 0$, $\gamma_c, \gamma_o \geq 0$, $v \in L^\infty(0, T; L^\infty(\Omega; \mathbb{R}^d))$ with $d \in \{1, 2, 3\}$ and $\nabla \cdot v = 0$, $y_{out} \in L^s(0, T)$, $s > d + 1$, $y_o \in C(\overline{\Omega})$ and $b_1, \ldots, b_m \in L^\infty(\Gamma_c)$. 
Remark 1.2. The parameter $\lambda > 0$ represents the diffusion coefficient and $v$ is the velocity field in the room. We assume that the velocity field solves the incompressible Navier-Stokes equation. Therefore, it holds naturally that $\nabla \cdot v = 0$. Moreover, the choice of the Robin boundary conditions is justified by the so-called Newton’s law of cooling, simplification of the non-linear Stefan-Boltzmann law for radiation. These simplified conditions are used sometimes in engineering’s models for room heating, e.g., [6, 131]. Thanks to the constants $\gamma_c$ and $\gamma_o$ we can model in an approximated manner the heat transfer fluxes between the heat sources and the room and how the outside temperature $y_{out}$ influences the temperature $y$, respectively.

Furthermore, the assumption on the regularity of the temperature $y_{out}$ is justified for the result in Section 1.1.1, where we can prove continuity for the weak solution of the state equation. This assumption is not restrictive: since $T$ is bounded we have $L^s((0,T) \subset L^2(0,T)$ for $s > d + 1 \geq 2$ with $d \in \{1, 2, 3\}$. ♦

To write (1.1) in weak form we introduce the non-symmetric, time-dependent bilinear form $a(t; \cdot, \cdot) : V \times V \to \mathbb{R}$

$$a(t; \varphi, \phi) = \int_{\Omega} \lambda \nabla \varphi \cdot \nabla \phi + (v(t) \cdot \nabla \varphi) \phi \, dx + \gamma_c \int_{\Gamma_c} \varphi \phi \, ds + \gamma_o \int_{\Gamma_o} \varphi \phi \, ds$$

for $\varphi, \phi \in V$ a.e. in $[0, T]$ and the time-dependent linear functional $F(t) : V \to V'$

$$\langle F(t), \varphi \rangle_{V', V} = \gamma_o y_{out}(t) \int_{\Gamma_o} \varphi \, ds \quad \text{for } \varphi \in V \text{ a.e. in } [0, T],$$

where $\langle \cdot, \cdot \rangle_{V', V}$ stands for the dual pairing between $V$ and its dual space $V'$. Moreover, the linear operator $B : \mathbb{R}^m \to V'$ is defined as

$$\langle Bu, \varphi \rangle_{V', V} = \gamma_c \sum_{i=1}^{m} u_i \int_{\Gamma_c} b_i \varphi \, ds \quad \text{for all } \varphi \in V$$

for given $u = (u_i)_{1 \leq i \leq m} \in \mathbb{R}^m$. The dual operator $B^* : V \to \mathbb{R}^m$ of $B$ satisfies

$$\langle B^* \varphi, u \rangle_{\mathbb{R}^m} = \langle Bu, \varphi \rangle_{V', V} = \gamma_c \sum_{i=1}^{m} u_i \int_{\Gamma_c} b_i(s) \varphi(s) \, ds \quad (1.2)$$

for all $(u, \varphi) \in \mathbb{R}^m \times V$. Thus,

$$B^* \varphi = \left( \begin{array}{c} \gamma_c \int_{\Gamma_c} b_1(s) \varphi(s) \, ds \\ \vdots \\ \gamma_c \int_{\Gamma_c} b_m(s) \varphi(s) \, ds \end{array} \right) \in \mathbb{R}^m \quad \text{for } \varphi \in V.$$
Definition 1.3. The state variable \( y \in W(0, T) \) is called a weak solution to (1.1) if
\[
\frac{d}{dt} \langle y(t), \varphi \rangle_H + a(t; y(t), \varphi) = (F(t) + B(u(t)), \varphi) \quad \text{for all } \varphi \in V \text{ a.e. in } (0, T],
\]
\[
y(0) = y_0 \quad \text{in } H
\]
is satisfied.

Lemma 1.4. Let Assumption 1.1 hold. Then:
1) For almost all \( t \in [0, T] \) the bilinear form satisfies
\[
|a(t; \varphi, \phi)| \leq \alpha \|\varphi\|_V \|\phi\|_V \quad \text{for all } \varphi, \phi \in V,
\]
\[
a(t; \varphi, \varphi) \geq \alpha_1 \|\varphi\|_V^2 - \alpha_2 \|\varphi\|_H^2 \quad \text{for all } \varphi \in V
\]
for constants \( \alpha, \alpha_1 > 0 \) and \( \alpha_2 \geq 0 \).
2) The mapping \( F \) belongs to \( L^2(0, T; V') \), i.e.,
\[
\|F\|_{L^2(0, T; V')} = \sup_{\|\varphi\|_{L^2(0, T; V')} = 1} \int_0^T \langle F(t), \varphi(t) \rangle_{V', V} dt \leq c_F.
\]
for a constant \( c_F > 0 \).
3) The linear operator \( B \) is bounded, i.e., there is a constant \( c_B > 0 \) such that
\[
\|B\|_{L(\mathbb{R}^m, V')} = \sup_{|u|_2 = 1} \|Bu\|_{V'} \leq c_B,
\]
where \( |\cdot|_2 \) denotes the Euclidean norm (here in \( \mathbb{R}^m \)).

Proof. 1) By the trace theorem [35, p. 258] there exists a constant \( c_\Gamma > 0 \) such that
\[
\max \left\{ \|\phi\|_{L^2(\Gamma)}, \|\phi\|_{L^2(\Gamma_\varepsilon)} \right\} \leq \|\phi\|_{L^2(\Gamma)} \leq c_\Gamma \|\phi\|_V \quad \text{for all } \phi \in V. \quad (1.4)
\]
Then, the first bound follows from
\[
|a(t; \varphi, \phi)| \leq \lambda \int_\Omega |\nabla \varphi \cdot \nabla \phi| \, dx + \|v(t)\|_{L^\infty(\Omega, \mathbb{R}^d)} \int_\Omega |\nabla \varphi|_2 \|\phi\|_2 \, dx
\]
\[
+ \gamma_c \int_{\Gamma_\varepsilon} |\varphi| \|\phi\|_H \, ds + \gamma_0 \int_{\Gamma_\varepsilon} |\varphi||\phi|_H \, ds
\]
\[
\leq \lambda \|\varphi\|_V \|\phi\|_V + \|v\|_{L^\infty(0, T; L^\infty(\Omega, \mathbb{R}^d))} \|\varphi\|_V \|\phi\|_V
\]
\[
+ (\gamma_c + \gamma_0) c_\Gamma^2 \|\varphi\|_V \|\phi\|_V
\]
\[
\leq \alpha \|\varphi\|_V \|\phi\|_V
\]
for every \( \varphi, \phi \in V \) with \( \alpha = \lambda + (\gamma_c + \gamma_o) c_T^2 + \| \mathbf{v} \|_{L^\infty(0,T;L^\infty(\Omega;\mathbb{R}^d))} \). Using Young’s inequality [4, p. 28]

\[
ab \leq \frac{\epsilon a^2}{2} + \frac{b^2}{2\epsilon} \quad \text{for } a, b \geq 0 \text{ and } \epsilon > 0
\]

we have

\[
a(t; \varphi, \varphi) \geq \lambda \left( \| \varphi \|_V^2 - \| \varphi \|_H^2 \right) - \| \mathbf{v} \|_{L^\infty(0,T;L^\infty(\Omega;\mathbb{R}^d))} \| \varphi \|_V \| \varphi \|_H
\]

\[
\geq \lambda \left( \| \varphi \|_V^2 - \| \varphi \|_H^2 \right) - \frac{\lambda}{2} \| \varphi \|_V^2 - \frac{\| \mathbf{v} \|_{L^2(0,T;L^\infty(\Omega;\mathbb{R}^d))}^2}{2\lambda} \| \varphi \|_H^2
\]

\[
= \alpha_1 \| \varphi \|_V^2 - \alpha_2 \| \varphi \|_H^2
\]

with \( \alpha_1 = \lambda/2 \) and \( \alpha_2 = \lambda + \| \mathbf{v} \|_{L^\infty(0,T;L^\infty(\Omega;\mathbb{R}^d))}^2/(2\lambda) \).

2) For every \( \varphi \in L^2(0,T;V) \) we derive that

\[
\int_0^T \langle \mathcal{F}(t), \varphi(t) \rangle_{V',V} dt = \int_0^T \gamma_o y_{out}(t) \int_{\Gamma_o} \varphi(t) \, d\mathbf{s} \, dt
\]

\[
\leq \int_0^T c_1 |y_{out}(t)||\varphi(t)||_{L^2(\Gamma_o)} \, dt,
\]

where we set \( c_1 = \gamma_o(\int_{\Gamma_o} \, d\mathbf{s})^{1/2} \geq 0 \). Using (1.4), setting \( c_2 = c_1 c_T \) and considering the Cauchy-Schwarz inequality we conclude that

\[
\int_0^T \langle \mathcal{F}(t), \varphi(t) \rangle_{V',V} dt \leq c_F \| \varphi \|_{L^2(0,T;V)}
\]

for every \( \varphi \in L^2(0,T;V) \) with \( c_F = c_2 \| y_{out} \|_{L^2(0,T)} \). Now, the claim follows directly.

3) From (1.4) we infer that

\[
\| \mathbf{B} u \|_{V'} = \gamma_c \sup_{\| \varphi \|_V = 1} \sum_{i=1}^m u_i \int_{\Gamma_c} b_i \varphi(t) \, d\mathbf{s} \leq \gamma_c c_T \sum_{i=1}^m \| b_i \|_{L^2(\Gamma_c)} \sup_{\| \varphi \|_V = 1} |u_i||\varphi||_V
\]

\[
\leq \gamma_c c_T \sum_{i=1}^m \| b_i \|_{L^2(\Gamma_c)} |u_i| \leq c_B |u|_2
\]

with \( c_B = \gamma_c c_T (\sum_{i=1}^m \| b_i \|_{L^2(\Gamma_c)}^2)^{1/2} \geq 0 \), which gives the desired estimate. \( \square \)
**Proposition 1.5.** Suppose that Assumption 1.1 is satisfied. Then, (1.3) possesses a unique solution \( y \in W(0,T) \) for every \( u \in \mathcal{U}_{ad} \) and

\[
\|y\|_{W(0,T)} \leq c_y (\|y_0\|_H + \|y_{out}\|_{L^2(0,T)} + \|u\|_U)
\]

for a constant \( c_y \geq 0 \) which is independent of \( y_0, y_{out} \) and \( u \).

**Proof.** Existence of a unique solution to (1.3) follows directly from Lemma 1.4 and [29, pp. 512-520]. Moreover, since \( T \) is bounded and \( y_{out} \in L^s(0,T) \) for \( s > d + 1 \), \( d \in \{1, 2, 3\} \), this implies that \( y_{out} \in L^2(0,T) \), then the a-priori bound is shown in [126, Theorem 3.13].

**Remark 1.6.** We split the solution to (1.3) in one part, which depends on the fixed initial condition \( y_0 \) and on the outside temperature \( y_{out} \), and another part depending linearly on the control variable. Let \( \hat{y} \in W(0,T) \) be the unique solution to the problem

\[
\frac{d}{dt} \langle \hat{y}(t), \varphi \rangle_H + a(t; \hat{y}(t), \varphi) = \langle F(t), \varphi \rangle_{V',V} \quad \text{for all } \varphi \in V \text{ a.e. in } (0,T],
\]

\[
\hat{y}(0) = y_0 \quad \text{in } H.
\]

We define the subspace

\[
W_0(0,T) = \{ \varphi \in W(0,T) \mid \varphi(0) = 0 \text{ in } H \}
\]

endowed with the topology of \( W(0,T) \). Let us now introduce the linear solution operator, known as control-to-state map, \( S : \mathcal{U} \to W_0(0,T) \): for \( u \in \mathcal{U} \) the function \( y = Su \in W_0(0,T) \) is the unique solution to

\[
\frac{d}{dt} \langle y(t), \varphi \rangle_H + a(t; y(t), \varphi) = \langle (B(u(t)), \varphi) \rangle_{V',V} \quad \text{for all } \varphi \in V \text{ a.e. in } (0,T].
\]

From \( y \in W_0(0,T) \) it follows that \( y(0) = 0 \) in \( H \). The boundedness of \( S \) follows from (1.5). Now, the solution to (1.3) can be expressed as \( y = \hat{y} + Su \).

**Remark 1.7.** To simplify the notation, throughout the thesis, we will not distinguish between the control operator \( B : \mathbb{R}^m \to V' \) and the operator \( \tilde{B} : \mathcal{U} \to L^2(0,T;V') \), where

\[
\langle \tilde{B}u, \varphi \rangle_{L^2(0,T;V'),L^2(0,T;V)} := \int_0^T \langle B(u(t)), \varphi(t) \rangle_{V',V} dt \quad \text{for all } \varphi \in L^2(0,T;V).
\]

We indicate both operators with the notation \( B \) and their adjoint operators with \( B^* \).
1.1 The state equation

1.1.1 Maximal Regularity

For later proofs, we want to extend the proof of Lemma 7.12 in [126, pp. 378] to our setting, where we have different boundary conditions and an additional advection term. Therefore, we want to prove the following result:

**Lemma 1.8.** Let $\Omega \subset \mathbb{R}^d$ be a bounded domain for $d \in \{1, 2, 3\}$. Suppose that $f \in L^r(Q)$, $g^1 \in L^s(\Sigma_c)$, $g^2 \in L^s(\Sigma_o)$ and $y_o \in C(\overline{\Omega})$ are given functions with $r > d/2 + 1$ and $s > d + 1$. Moreover, let Assumption 1.1 be satisfied, then the weak solution $y$ to

$$y(t, x) - \lambda \Delta y(t, x) + \nu(t, x) \cdot \nabla y(t, x) = f(t, x), \quad \text{a.e. in } Q,$$

$$\lambda \frac{\partial y}{\partial n}(t, s) + \gamma_c y(t, s) = \gamma_c g^1(t, s), \quad \text{a.e. on } \Sigma_c,$$

$$\lambda \frac{\partial y}{\partial n}(t, s) + \gamma_o y(t, s) = \gamma_o g^2(t, s), \quad \text{a.e. on } \Sigma_o,$$

$$y(0, x) = y_o(x), \quad \text{a.e. in } \Omega$$

belongs to $W(0, T) \cap C(\overline{\Omega})$. Furthermore, there exists a constant $c = c(r, s) > 0$, which does not depend on $f$, $g^1$ and $g^2$, such that

$$\|y\|_{W(0, T)} + \|y\|_{C(\overline{\Omega})} \leq c(r, s) \left(\|f\|_{L^r(Q)} + \|g^1\|_{L^s(\Sigma_c)} + \|g^2\|_{L^s(\Sigma_o)}\right)$$

At last, let $\{f_k\}_{k \in \mathbb{N}} \subset L^r(Q)$, $\{g^1_k\}_{k \in \mathbb{N}} \subset L^s(\Sigma_c)$ and $\{g^2_k\}_{k \in \mathbb{N}} \subset L^s(\Sigma_o)$ be three sequences converging weakly to $f \in L^r(Q)$, $g^1 \in L^s(\Sigma_c)$ and $g^2 \in L^s(\Sigma_o)$ respectively, then $y_k$, the weak solution of (1.6) with data $f_k$, $g^1_k$ and $g^2_k$, converges strongly in $L^2(0, T; H)$ and in $C(\overline{\Omega})$ to $y$, the weak solution of (1.6) with data $f$, $g^1$ and $g^2$.

**Proof.** As also stated in [126], for $y_o \equiv 0$ the proof relies on the more general results contained in [41, 42]. What we have to do is essentially to follow step by step the proof in [126] and check if those results can be applied to our setting. Since $y(0, x) = 0$ a.e. in $\Omega$, the result follows from Theorem 7.2 in [42]. Anyway, we check if our problem fulfills the required assumptions. In [42], the author deals with equation of the type $y' + Ay + By = F$, where $A : L^2(0, T; V) \to L^2(0, T; V')$ is of the form

$$\langle Ay, \varphi \rangle_{L^2(0, T; V')} = \int_0^T \int_\Omega \nabla y(t) \cdot \nabla \varphi(t) \, dx \, dt \quad \text{for } y, \varphi \in L^2(0, T; V).$$

Furthermore, applying the divergence theorem and using the fact that $\nabla \cdot \nu = 0$, one can obtain

$$\int_\Omega \nu \cdot \nabla y(t) \varphi(t) \, dx = -\int_\Omega y(t) \nu(t) \cdot \nabla \varphi(t) \, dx + \int_\Gamma (\nu(t) \cdot n)y(t)\varphi(t) \, ds.$$
1. The linear-quadratic optimal control problem

Therefore, in our setting the operator $\mathcal{B}$ in [42] can be expressed as

$$
(\mathcal{B}y, \varphi) = \int_0^T \int_\Omega y(t)(-v(t) \cdot \nabla \varphi(t)) \, dx + \int_\Gamma (v(t) \cdot n + \gamma(t))y(t)\varphi(t) \, ds,
$$

where $\gamma(t, s) := \gamma_\ell \chi_{\Sigma_\ell}(t, s) + \gamma_\sigma \chi_{\Sigma_\sigma}(t, s)$ for almost all $s \in \Gamma$ and $t \in (0, T)$. Thus, choosing $b(t) = -v(t)$ and $b_\Gamma(t) = v(t) \cdot n + \gamma(t)$ for almost all $t \in (0, T)$. Since $b_\Gamma \in L^\infty(0, T; L^\infty(\Omega))$ and $b \in L^\infty(0, T; L^\infty(\Omega; \mathbb{R}^d))$ for Assumption 1.1, we can apply Theorem 7.2 in [42]. Accordingly, there exists a $\tilde{\omega} \in (d, d + 2]$ such that for any $\omega \in [0, \tilde{\omega})$ the restriction of the operator $\mathcal{P} : y \rightarrow y' + Ay + B y$ to the space $W^\omega_0(0, T) := \{ \varphi \in W^\omega(0, T) : \varphi(0) = 0 \}$ is a linear isomorphism between the space $W^\omega_0(0, T)$ and $L^2_2(0, T; V')$. These spaces are called Sobolev-Morrey spaces and a detailed explanation of such spaces and their properties can be found in [41]. In this work, we do not address all the details to keep the proof as simple as possible, recalling only the properties and the definitions we need. At first, in [41, pp. 802] the Morrey space $L^\infty_2(0, T; H)$ is defined as the set of all $\varphi \in L^2(0, T; H)$ that such that

$$
[\varphi]^2_{L^\infty_2(0, T; H)} := \sup_{(I, G) \in T_r \times \Omega_r, r > 0} r^{-\omega} \int_I \int_G |\varphi(t, x)|^2 \, dx \, dt < +\infty,
$$

where $T_r := \{(0, t) \cap (t - r^2, t) | t \in (0, T)\}$ and $\Omega_r := \{ \Omega \cap C_r(x) \mid x \in \Omega \}$ with $C_r(x) := \{ z \in \mathbb{R}^d : \|z - x\|_\infty < r \}$ the open cube of radius $r > 0$ centered in $x \in \mathbb{R}^d$. This space is a Banach space endowed with the norm

$$
\|\varphi\|^2_{L^\infty_2(0, T; H)} = \|\varphi\|^2_{L^2(0, T; H)} + [\varphi]^2_{L^\infty_2(0, T; H)}.
$$

In a similar way, one can define $L^2_2(0, T; L^2(\Gamma))$. Moreover

$$
L^2_2(0, T; V) := \{ \varphi \in L^2(0, T; V) \mid \varphi \in L^\infty_2(0, T; H), |\nabla \varphi| \in L^\infty_2(0, T; H) \}.
$$

In [41, pp. 815], one can also find the definition of the Sobolev-Morrey space $L^\omega_2(0, T; V')$. In the end, taking in account the duality mapping from $V$ to $V'$, the Sobolev-Morrey space $W^\omega(0, T)$ is defined as

$$
W^\omega(0, T) := \{ \varphi \in L^\omega_2(0, T; V) \mid \varphi_t \in L^\omega_2(0, T; V') \}
$$

with the choice $X = Y = V$ in Definition 6.1 of [41], as also stated in [126]. The property of $W^\omega(0, T)$ we want to use follows by Theorem 3.4 and Theorem 6.8 in [41], where it is shown that $W^\omega(0, T)$ is continuously embedded in the space of Hölder continuous functions $C([0, T]; C^{0, \alpha}(\overline{\Omega})) \cap C^{0, \alpha/2}([0, T]; C(\overline{\Omega}))$ for $\omega \in (d, d + 2]$ and
\[ \alpha = (\omega - d)/2 \]  
Since \( \alpha > 0 \) for all \( \omega \in (d, d + 2) \) we have therefore continuity
up to the boundary of \( Q \). What remains to prove is under which conditions on \( r \)
and \( s \) the mapping \((f, g^1, g^2) \rightarrow y\) is continuous from \( L^r(Q) \times L^s(\Sigma_c) \times L^s(\Sigma_\omega) \to W^{\omega}(0, T; V)\) for
some \( \omega \in (d, \bar{\omega}) \) with \( \bar{\omega} \in (d, d + 2) \). Theorem 5.6 in [41] states
that for \( \omega \in [0, d + 2] \) the map \((h, f, g) \rightarrow F(h, f, g) \in V'\) defined by
\[
(F, \varphi)_{V', V} = \int_0^T \int_O (h(t) \cdot \nabla \varphi(t) + f(t) \varphi(t)) \, dx \, dt + \int_0^T \int_{\Gamma} g(t) \varphi(t) \, ds \, dt
\]
is a bounded linear operator from \( L^r_2(0, T; L^2(\Omega; \mathbb{R}^d)) \times L^{r_2-2}(0, T; H) \times L^{r_2-1}(0, T; L^2(\Gamma)) \) to \( V' \). In our case \( g(t, s) = \gamma_c g^1(t, s) \chi_{\Sigma_c}(t, s) + \gamma_\omega g^2 \chi_{\Sigma_\omega}(t, s) \) for
almost all \( s \in \Gamma \) and \( t \in (0, T) \), which is in \( L^s(\Sigma) \) for \( s > d + 1 \geq 2 \). Following
Remark 3.4 and 3.7 in [41] we have

- If \( r \geq 2 \) and \( \omega_r - 2 = d \left(1 - \frac{2}{r} \right) + 2 \left(1 - \frac{2}{r_2} \right) \in [0, d + 2] \), then \( L^r(\omega_r) \) is
continuously embedded in \( L^{\omega_r-2}(0, T; H) \). Since we want \( \omega_r \in (d, \bar{\omega}) \), we need that \( r > \frac{d}{2} + 1 \).

- If \( s \geq 2 \) and \( \omega_s - 1 = (d - 1) \left(1 - \frac{2}{s} \right) + 2 \left(1 - \frac{2}{s_2} \right) \in [0, d + 1] \), then \( L^s(\omega_s) \) is
continuously embedded in \( L^{\omega_s-1}(0, T; L^2(\Gamma)) \). Since we want \( \omega_s \in (d, \bar{\omega}) \), we need that \( s > d + 1 \).

These conditions are exactly the one we have imposed for the right-hand side and
the boundary data. Therefore we have proved the claim. A different proof, but for
the same problem with \( y_0 \neq 0 \) can be found in [27, pp. 1306-1307]. We refer to [27]
to complete the Lemma’s proof, but without going in technical details. Moreover,
the a-priori bound can be shown following [126, Theorem 3.13]. In the end, applying
Aubin-Lions Lemma we have that \( W(0, T) \) is compactly embedded in \( L^2(0, T; H) \);
see [123] and references therein. Moreover, \( W^{\omega}(0, T) \) is continuously embedded in
\( C([0, T]; C^{0,\alpha}(\overline{Q})) \cap C^{0,\alpha/2}([0, T]; C(\overline{Q})) \), which is compactly embedded in \( C(\overline{Q}) \)
for the Arzelà-Ascoli Theorem; see, e.g., [4]. Thus, we have the strong convergence
of \( y_k \) to \( y \) in \( L^2(0, T; H) \) and in \( C(\overline{Q}) \) if \( \{f_k\}_{k \in \mathbb{N}}, \{g^1_k\}_{k \in \mathbb{N}} \) and \( \{g^2_k\}_{k \in \mathbb{N}} \) converge weakly
to \( f, g^1 \) and \( g^2 \) in \( L^s(Q) \), \( L^s(\Sigma_c) \) and \( L^s(\Sigma_\omega) \), respectively. \( \square \)

**Remark 1.9.** As also observed in [126], the subspaces of \( H^1(\Omega) \) containing all
the functions which vanish on the Dirichlet boundary is indicated with \( H^1_0(\Omega) \) in
[42], where \( G = \Omega \cup \Gamma \) and \( \Gamma \subset \partial \Omega \) is the complement of the boundary where
Neumann or Robin boundary conditions can be prescribed. In our case \( \Gamma \equiv \partial \Omega \),
therefore the Dirichlet boundary is empty and we have that \( H^1_0(\Omega) = H^1(\Omega) = V \)
and \( H^{-1}(G) = H^1(\Omega)' = V' \). \( \diamond \)
Remark 1.10. As already mentioned, the assumption $\nabla \cdot v = 0$ in Assumption 1.1 (and therefore in Lemma 1.8) is due to our application setting. Lemma 1.8 can be formulated in a more general way replacing this assumption with the more general one $\nabla \cdot v \in L^\infty(0,T;L^\infty(\Omega))$, which does not change the proof of the result.

As consequence of Lemma 1.8, we have the following property of high regularity for the weak solution of (1.1):

Corollary 1.11. Let Assumption 1.1 hold and let $u \in U_{ad}$, then $y = \hat{y} + Su \in W(0,T) \cap C(Q)$. Moreover, the estimate

$$
\|y\|_{W(0,T)} + \|y\|_{C(Q)} \leq c_y(s)(\|y_0\|_H + \|y_{out}\|_{L^\infty(0,T)} + \|u\|_{L^\infty(0,T;\mathbb{R}^m)})
$$

(1.14)

holds true for $s > d+1$ and $c_y(s) \geq 0$, which is independent of $y_0$, $y_{out}$ and $u$.

Proof. For Assumption 1.1, $y_0 \in C(\bar{\Omega})$ and $\gamma_0y_{out} \in L^s(\Sigma_0)$ for $s > d+1$. Moreover, since $T$ is bounded and $u_{ai}(t) \leq u_i(t) \leq u_{bi}(t)$ for $i = 1, \ldots, m$ and a.e. in $[0,T]$, we have that $u \in L^\infty(0,T;\mathbb{R}^m)$ and therefore it is in $L^s(0,T;\mathbb{R}^m)$ for $s \in [1, +\infty]$, in particular for $s > d+1$. Thus, $\gamma_c \sum_{i=1}^{m} b_i u_i \in L^s(\Sigma_c)$ and applying Lemma 1.8 we have the claim.

1.2 The state-constrained optimal control problem

In our application the temperature $y$ has to satisfy lower and upper bounds. Therefore, we consider the following optimization problem:

$$
\min J(y, u) := \frac{\sigma_Q}{2} \int_0^T \|y(t) - y_Q(t)\|_H^2 \, dt + \frac{\sigma_T}{2} \|y(T) - y_T\|_H^2
$$

$$
+ \frac{1}{2} \sum_{i=1}^{m} \sigma_i \|u_i\|_{L^2(0,T)}^2
$$

(1.15a)

subject to the state equations (1.1) and to the inequality constraints

$$
u_{ai}(t) \leq u_i(t) \leq u_{bi}(t) \quad \text{a.e. in } [0,T] \text{ for } i = 1, \ldots, m,
$$

$$y_{a}(t,x) \leq y(t,x) \leq y_{b}(t,x) \quad \text{a.e. in } Q_T := (0,T] \times \Omega.
$$

(1.15b)

(1.15c)

We suppose the following properties of the parameters in (1.15):
1.2 The state-constrained optimal control problem

Assumption 1.12. \( y_Q \in L^2(0,T;H), \ y_T \in H, \ \sigma_Q, \ \sigma_T \geq 0, \ \sigma_1, \ldots, \sigma_m > 0 \) and \( y_a, y_b \in C(\bar{Q}) \).

Remark 1.13. The parameters \( y_Q \) and \( y_T \) are desired temperature states. Furthermore, \( y_a \) and \( y_b \) are given pointwise lower and upper bounds for the temperature variable, respectively. Moreover, the control constraints \( u_a \) and \( u_b \) represent physical restrictions on the heating devices. From (1.15b) and (1.15c), it holds clearly that \( u_a(t) \leq u_b(t) \) a.e. in \([0,T]\) and \( y_a(t,x) \leq y_b(t,x) \) a.e. in \( Q_T \), respectively.

♦

Considering Remark 1.6, we can write the previous problem as a pure control constrained optimization problem. We define \( \hat{y}_Q(t) := y_Q(t) - \hat{y}, \ \hat{y}_T := y_T - \hat{y}(T), \ \hat{y}_a := y_a - \hat{y} \) and \( \hat{y}_b := y_b - \hat{y} \), thus considering the control-to-state map \( S \) introduced in Remark 1.6 we define the reduced cost functional

\[
\hat{J}(u) := J(y(u), u) = \frac{\sigma_Q}{2} \int_0^T \| (Su)(t) - \hat{y}_Q(t) \|_H^2 + \frac{\sigma_T}{2} \| (Su)(T) - \hat{y}_T \|_H^2 + \frac{1}{2} \sum_{i=1}^m \sigma_i \| u_i \|_{L^2(0,T)}^2
\]

and the admissible set

\[
\hat{U}_{ad} = \{ u \in \mathcal{U} \ | \ u_a(t) \leq u_i(t) \leq u_b(t), i = 1, \ldots, m \text{ and a.e. in } [0,T], \ \hat{y}_a \leq Su \leq \hat{y}_b \text{ a.e. in } Q_T \}.
\]

Thus, the optimal control problem (1.15) is equivalent to the problem

\[
\min \hat{J}(u) \quad \text{s.t.} \quad u \in \hat{U}_{ad}. \quad (\hat{P})
\]

Assumption 1.14. There exist \( \bar{u} \in \mathcal{U} \) such that \( u_a(t) \leq \bar{u}_i(t) \leq u_b(t) \) a.e. in \([0,T]\) for \( i = 1, \ldots, m \) and there exist \( \tau > 0 \) such that \( \hat{y}_a + \tau \leq S\bar{u} \leq \hat{y}_b - \tau \) a.e in \( Q_T \), i.e. \( \hat{U}_{ad} \) has a feasible point.

Theorem 1.15 (Existence of unique solution). Let Assumptions 1.1, 1.12 and 1.14 hold. Then, there exists a unique solution \( \bar{u} \in \hat{U}_{ad} \) of \((\hat{P})\).

Proof. The set \( \hat{U}_{ad} \) is closed, convex and non-empty, which means that problem \((\hat{P})\) has a feasible point. Moreover, \( \sigma_i > 0 \) for all \( i = 1, \ldots, m \), therefore we can apply Theorem 1.43 in [67, pp.53-54] that completes the proof.

Corollary 1.16. Let Assumptions 1.1, 1.12 and 1.14 hold. Then, there exists a unique solution \((\bar{y}, \bar{u}) \in W(0,T) \times U_{ad} \) of (1.15).
1. The linear-quadratic optimal control problem

1.2.1 First-order optimality conditions

In order to derive optimality conditions for problem $\hat{P}$, we need to introduce the following space of Radon measures:

**Definition 1.17.** Let $Q \subset \mathbb{R}^{d+1}$. The space $\mathcal{M}(Q)$ of Radon measures is defined as the dual space of $C(Q)$ and it is endowed with the norm

$$
\|\mu\|_{\mathcal{M}(Q)} = \sup_{f \in C(Q), \|f\|_{C(Q)} \leq 1} \int_Q f d\mu
$$

(1.16)

Now, the first-order optimality conditions for $\hat{P}$ can be resumed in the following result

**Theorem 1.18.** Let Assumptions 1.1, 1.12 and 1.14 hold. Let $\bar{u} \in \hat{U}_{ad}$ be the optimal solution $\hat{P}$ with associated optimal state $\bar{y} = \hat{y} + \mathcal{S}\bar{u}$. Then, there exist unique Lagrange multipliers $\bar{\mu}_a, \bar{\mu}_b \in \mathcal{M}(Q)$, $\bar{p} \in L^r(0,T;W^{1,s}(\Omega))$ for all $r,s \in [1,2)$ with $\frac{2}{r} + \frac{d}{s} > d + 1$ and $\bar{\alpha} = (\bar{\alpha}_i)_{1 \leq i \leq m} \in \mathcal{U}$ satisfying the dual equations

$$
\int_0^T \frac{d}{dt} \langle \varphi(t), \bar{p}(t) \rangle_H + a(t; \varphi(t), \bar{p}(t)) - \sigma_Q \langle y_Q(t) - \bar{y}(t), \varphi(t) \rangle_H \, dt = \int_Q \varphi \, d\bar{\mu}_a - \int_Q \varphi \, d\bar{\mu}_b
$$

(1.17a)

for all $\varphi \in W_0^\infty(0,T)$, where

$$
W_0^\infty(0,T) := \{ \varphi \in W(0,T) \cap C(\overline{Q}) \mid \varphi(0,\cdot) = 0 \text{ in } \overline{Q}, \varphi_t - \lambda \Delta \varphi + v \cdot \nabla \varphi \in L^\infty(\overline{Q}) \}.
$$

Furthermore, the Lagrange multipliers satisfy the optimality system

$$
\sigma_i \bar{u}_i - \gamma_i \int_{T_i} b_i \bar{p} \, ds + \bar{\alpha}_i = 0 \text{ in } L^2(0,T) \text{ for } i = 1, \ldots, m,
$$

(1.17b)

$$
\bar{\mu}_a \geq 0, \quad \bar{y}(t,\mathbf{x}) \geq y_a(t,\mathbf{x}) \text{ a.e. in } Q_T \quad \text{and} \quad \int_Q (\bar{y} - y_a) \, d\bar{\mu}_a = 0,
$$

(1.17c)

$$
\bar{\mu}_b \geq 0, \quad \bar{y}(t,\mathbf{x}) \leq y_b(t,\mathbf{x}) \text{ a.e. in } Q_T \quad \text{and} \quad \int_Q (y_b - \bar{y}) \, d\bar{\mu}_b = 0
$$

(1.17d)

and the complementarity condition

$$
\bar{\alpha}_i = \max \{ 0, \bar{\alpha}_i + \eta_i (\bar{u}_i - u_{b_i}) \} + \min \{ 0, \bar{\alpha}_i + \eta_i (\bar{u}_i - u_{a_i}) \}
$$

(1.17e)

for $i = 1, \ldots, m$ and for arbitrarily chosen $\eta_1, \ldots, \eta_m > 0$, where the max- and min-operations are interpreted componentwise in the pointwise everywhere sense.
Proof. The proof of this result can be obtained following, e.g., [27, 39, 67, 103]. ☐

Remark 1.19. a) Since \( \bar{\mu}_a \) and \( \bar{\mu}_b \) are measures, we need to clarify the meaning of \( \mu \geq 0 \) for \( \mu \in \mathcal{M}(\bar{Q}) \). This expression is equivalent to \( \langle \mu, \varphi \rangle_{\mathcal{M}(\bar{Q}), C(\bar{Q})} \geq 0 \) for all \( \varphi \in C(\bar{Q}) \) with \( \varphi(t, x) \geq 0 \) for all \( (t, x) \in \bar{Q} \).

b) The condition (1.17e) is a nonlinear complementarity problem (NCP) function based reformulation of the complementarity system

\[
\bar{\alpha}_a \geq 0, \quad u_a - \bar{u} \leq 0, \quad \langle \bar{\alpha}_a, u_a - \bar{u} \rangle_U = 0, \\
\bar{\alpha}_b \geq 0, \quad \bar{u} - u_b \leq 0, \quad \langle \bar{\alpha}_b, \bar{u} - u_b \rangle_U = 0
\]

with \( \bar{\alpha} = \bar{\alpha}_b - \bar{\alpha}_a \in \mathcal{U} \). ☐

Remark 1.20. Analogously to Remark 1.6 we split the adjoint variable \( p \) into one part depending on the fixed desired states and into two other parts, which depend linearly on the control variable and on the measures \( \bar{\mu}_a \) and \( \bar{\mu}_b \), respectively. Let \( \hat{p} \in W(0, T) \) denote the unique solution to the adjoint equation

\[
-\frac{d}{dt} \langle \hat{p}(t), \varphi \rangle_H + a(t; \varphi, \hat{p}(t)) = \sigma_Q \langle \hat{y}_Q(t), \varphi \rangle_H \quad \text{for all } \varphi \in V \text{ a.e. in } [0, T], \\
\hat{p}(T) = \sigma_T \hat{y}_T \quad \text{in } H,
\]

where \( \hat{y}_Q \) and \( \hat{y}_T \) are defined in Section 1.2. Further, we define the linear bounded operator \( A_1 : \mathcal{U} \rightarrow W(0, T) \) as follows: for given \( u \in \mathcal{U} \) the function \( p = A_1 u \) is the unique solution to

\[
-\frac{d}{dt} \langle p(t), \varphi \rangle_H + a(t; \varphi, p(t)) = -\sigma_Q \langle (Su)(t), \varphi \rangle_H \quad \text{for all } \varphi \in V \text{ a.e. in } [0, T], \\
p(T) = -\sigma_T (Su)(T) \quad \text{in } H.
\]

Moreover, the linear bounded operator \( A_2 : \mathcal{M}(\bar{Q}) \rightarrow L^r(0, T; W^{1,s}(\Omega)) \) for all \( r, s \in [1, 2) \) with \( \frac{2}{r} + \frac{d}{s} > d + 1 \) is defined in the following way: for given \( \mu \in \mathcal{M}(\bar{Q}) \) the function \( p = A_2 \mu \) uniquely solves

\[
\int_0^T -\frac{d}{dt} \langle \varphi(t), p(t) \rangle_H + a(t; \varphi(t), p(t)) \, dt = \int_{\bar{Q}} \varphi \, d\mu
\]

for all \( \varphi \in W_0^{\infty}(0, T) \). Thus, the weak solution to (1.17a) can be expressed as \( \bar{p} = \hat{p} + A_1 \bar{u} + A_2 (\bar{\mu}_a - \bar{\mu}_b) \). ☐
Due to the fact that the cost functional $J$ is strictly convex, the previous first-order optimality conditions are sufficient and necessary, therefore to find the optimal solution, one has to solve the optimality system (1.17). Unfortunately, the presence of the measures $\bar{\mu}_a$ and $\bar{\mu}_b$ does not make the computation easy. For this reason, one can think on regularization techniques or penalty methods. The most common regularization techniques are the Lavrentiev regularization (e.g. [99]), the virtual control approach (e.g. [83]) and the Moreau-Yosida regularization (e.g. [21, 102]). Another possibility is to apply the augmented Lagrangian method, which belongs to the class of penalty methods; cf., e.g., [20, 72, 76].

1.2.2 Gradient of the reduced cost functional

In this section, we derive the gradient of the reduced cost functional $\hat{J}(u)$, which will be used in later proofs. Suppose that we choose an arbitrary $u \in \hat{U}_{ad}$. Then we set $y = \hat{y} + Su$ and $p = \hat{p} + A_1u$. It follows by standard arguments that the reduced cost functional is Fréchet-differentiable. In particular, we have for any direction $u^\delta \in \mathcal{U}$

$$\hat{J}'(u)u^\delta = \hat{J}_u(u)u^\delta = J_y(y, u)Su^\delta + J_u(y, u)u^\delta. \quad (1.18)$$

Note that

$$J_u(y, u)u^\delta = \int_0^T \sum_{i=1}^m \sigma_i u_i(t)w_i^\delta(t) \, dt. \quad (1.19)$$

Moreover, setting $y^\delta = Su^\delta \in W_0(0, T)$ we derive

$$J_y(y, u)Su^\delta = \int_0^T \sigma_Q \langle y(t) - y_Q(t), y^\delta(t) \rangle_H \, dt + \sigma_T \langle y(T) - y_T, y^\delta(T) \rangle_H. \quad (1.20)$$

To proceed in the computation, we use the following lemma:

**Lemma 1.21.** Suppose that $u \in \hat{U}_{ad}$, $y = \hat{y} + Su$ and $p = \hat{p} + A_1u$. Moreover, $y^\delta = Su^\delta$ for $u^\delta \in \mathcal{U}$. Then,

$$\int_0^T \sigma_Q \langle (y - y_Q)(t), y^\delta(t) \rangle_H \, dt = \sigma_T \langle y_T - y(T), y^\delta(T) \rangle_H - \int_0^T \sum_{i=1}^m \gamma_c u_i^\delta(t) \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt.$$
1.2 The state-constrained optimal control problem

Proof. At first, we notice that \( p \) and \( y^\delta \) satisfy the evolution problems

\[
\int_0^T -\langle p_t(t), \varphi(t) \rangle_{V',V} + a(t; \varphi(t), p(t)) \, dt \\
= \int_0^T \sigma_Q \langle (y_Q - y)(t), \varphi(t) \rangle_H \, dt \quad \text{for all } \varphi \in W(0,T)
\]  

(1.21)

and

\[
\int_0^T \langle y^\delta_t(t), \varphi(t) \rangle_{V',V} + a(t; y^\delta(t), \varphi(t)) \, dt \\
= \gamma_c \int_0^T \sum_{i=1}^m u_i^\delta(t) \langle b_i, \varphi(t) \rangle_{L^2(\Gamma_c)} \, dt \quad \text{for all } \varphi \in W(0,T).
\]  

(1.22)

respectively. Thus, choosing \( \varphi = y^\delta \) in (1.21), \( \varphi = p \) in (1.22), applying integration by parts and considering \( p(T) = \sigma_T(y_T - y(T)) \) and \( y^\delta(0) = 0 \), we find that

\[
\int_0^T \sigma_Q \langle (y - y_Q)(t), y^\delta(t) \rangle_H \, dt \\
= \int_0^T \langle p_t(t), y^\delta(t) \rangle_{V',V} - a(t; y^\delta(t), p(t)) \, dt \\
= \int_0^T -\langle y^\delta_t(t), p(t) \rangle_{V',V} - a(t; y^\delta(t), p(t)) \, dt + \langle p(T), y^\delta(T) \rangle_H - \langle p(0), y^\delta(0) \rangle_H \\
= - \int_0^T \langle y^\delta(t), p(t) \rangle_{V',V} + a(t; y^\delta(t), p(t)) \, dt + \sigma_T \langle y_T - y(T), y^\delta(T) \rangle_H \\
= - \int_0^T \sum_{i=1}^m \gamma_c u_i^\delta(t) \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt + \sigma_T \langle y_T - y(T), y^\delta(T) \rangle_H
\]

which is the claim. \( \square \)

From Lemma 1.21, (1.20) and (1.2) we infer that

\[
J_y(y, u) \mathcal{S} u^\delta = - \int_0^T \sum_{i=1}^m \gamma_c u_i^\delta(t) \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt.
\]  

(1.23)

Combining (1.19) and (1.23) we get for the derivative \( J^*(u) \in \mathcal{U} \) at \( u \in \hat{\mathcal{U}}_{\text{ad}} \) the formula

\[
\hat{J}^*(u) u^\delta = \int_0^T \sum_{i=1}^m \left( \sigma_i u_i(t) - \gamma_c \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \right) u_i^\delta(t) \, dt
\]  

(1.24)
1. The linear-quadratic optimal control problem

for every $u^\delta \in \mathcal{U}$ with $p = \hat{p} + A_1 u$ and therefore the gradient $\nabla \hat{J}(u) \in \mathcal{U}$ – the Riesz representant of $\hat{J}'(u)$ – reads

$$\nabla \hat{J}(u)(\cdot) = (\sigma_i u_i(\cdot) - \gamma_c \langle b_i, p(\cdot) \rangle_{L^2(\Gamma_i)})_{1 \leq i \leq m}$$

with $p = \hat{p} + A_1 u$.

1.3 The virtual control approach

At first, we set $W = L^2(0, T; H) \simeq L^2(Q)$, identifying, throughout the thesis, the spaces $L^2(0, T; H)$ and $L^2(Q)$ and the dual $W'$ with $W$. Moreover, let $y \in W(0, T)$ be given and $\mathcal{E} : W(0, T) \to W$ the canonical linear and bounded embedding operator. To simplify the notation all through the thesis we indicate with $\mathcal{E}y$ the solution $y \in W(0, T)$ itself and its embedded value $\mathcal{E}y$ in $W$. Let $\varepsilon > 0$ be chosen and $w \in W$ an additional (artificial) control. Then, (1.15c) is replaced by the mixed constraints

$$\begin{align*}
y_a(t, x) &\leq y(t, x) + \varepsilon w(t, x) \leq y_b(t, x) \quad \text{a.e. in } Q_T.
\end{align*}$$

We introduce the Hilbert space

$$\mathcal{X} = W(0, T) \times \mathcal{U} \times W$$

endowed with the common product topology. The set of admissible solutions is given by

$$\mathcal{X}_{ad} = \{(x = (y, u, w) \in \mathcal{X} \mid y = \hat{y} + Su, y_a \leq y + \varepsilon w \leq y_b, w_a \leq w \leq w_b \text{ and } u \in \mathcal{U}_{ad}\}$$

where $w_a, w_b \in \mathbb{R}$ and $w_a < 0 < w_b$. Let us mention that, for this problem, we do not explicitly indicate the dependence on $\varepsilon$, in order to simplify the notation and avoid confusion with Section 1.3.3, where this dependence is indicated for the general virtual control approach. The quadratic cost functional $J : \mathcal{X} \to \mathbb{R}$ is given by

$$\begin{align*}
J(x) &= \frac{\sigma_Q}{2} \int_0^T \|y(t) - y_Q(t)\|_H^2 \, dt + \frac{\sigma_T}{2} \|y(T) - y_T\|_H^2 \\
&\quad + \frac{1}{2} \sum_{i=1}^m \sigma_i \|u_i\|_{L^2(0, T)}^2 + \frac{\sigma_w}{2} \|w\|_W^2
\end{align*}$$
for \( x = (y, u, w) \in \mathcal{X} \). Moreover, the optimal control problem is

\[
\min J(x) \quad \text{subject to (s.t.)} \quad x \in \mathcal{X}_{ad}.
\]

Also problem \((P)\) can be formulated as pure control constrained problem. Recalling that \( \hat{y}_a = y_a - \hat{y} \in \mathcal{W} \) and \( \hat{y}_b = y_b - \hat{y} \in \mathcal{W} \), (1.15c) can be formulated equivalently in the variables \( u \) and \( w \) as follows:

\[
\hat{y}_a(t, x) \leq (Su)(t, x) + \varepsilon w(t, x) \leq \hat{y}_b(t, x) \quad \text{a.e. in } Q_T.
\]

We define \( Z = \mathcal{U} \times \mathcal{W} \) and introduce the bounded and linear mapping \( \mathcal{T} : Z \rightarrow Z \)

\[
z = (u, w) \mapsto \mathcal{T}(z) = \begin{pmatrix} u \\ \varepsilon w + Su \end{pmatrix} = \begin{pmatrix} \mathcal{I}_u & 0 \\ S & \varepsilon \mathcal{I}_W \end{pmatrix} \begin{pmatrix} u \\ w \end{pmatrix},
\]

where \( \mathcal{I}_u : \mathcal{U} \rightarrow \mathcal{U} \) and \( \mathcal{I}_W : \mathcal{W} \rightarrow \mathcal{W} \) stand for the identity operators in \( \mathcal{U} \) and \( \mathcal{W} \), respectively. Notice that \( \mathcal{T} \) is invertible and \( \mathcal{T}^{-1} \) is explicitly given as

\[
\mathcal{T}^{-1}(u, w) = \begin{pmatrix} \mathcal{I}_u & 0 \\ -\varepsilon^{-1}S & \varepsilon^{-1}\mathcal{I}_W \end{pmatrix} \begin{pmatrix} u \\ \varepsilon w + Su \end{pmatrix} = \begin{pmatrix} u \\ \frac{1}{\varepsilon} (w - Su) \end{pmatrix}
\]

for all \( z = (u, w) \in Z \). In fact, we have

\[
\mathcal{T}^{-1}(\mathcal{T}(u, w)) = \begin{pmatrix} \mathcal{I}_u & 0 \\ -\varepsilon^{-1}S & \varepsilon^{-1}\mathcal{I}_W \end{pmatrix} \begin{pmatrix} u \\ \varepsilon w + Su \end{pmatrix} = \begin{pmatrix} u \\ w \end{pmatrix}
\]

for all \( z = (u, w) \in Z \) and vice versa. Furthermore, we find

\[
\mathcal{T}^* = \begin{pmatrix} \mathcal{I}_u & \mathcal{S}^* \\ 0 & \varepsilon \mathcal{I}_W \end{pmatrix} : Z \rightarrow Z,
\]

and

\[
\mathcal{T}^{*-} = \begin{pmatrix} \mathcal{I}_u & -\varepsilon^{-1}\mathcal{S}^* \\ 0 & \varepsilon^{-1}\mathcal{I}_W \end{pmatrix} : Z \rightarrow Z,
\]

where we set \( \mathcal{T}^{*-} = (\mathcal{T}^*)^{-1} \).

**Remark 1.22.** Due to the simplification of the notation omitting the embedding operator \( \mathcal{E} : \mathcal{W}(0,T) \rightarrow \mathcal{W} \), it is important to remark that we do not indicate the term \( \mathcal{E} \) in the definition of the operator \( \mathcal{T} \). For the sake of completeness, we report the operators \( \mathcal{T} \) and its adjoint \( \mathcal{T}^* \) including the omitted operator \( \mathcal{E} \):

\[
\mathcal{T} = \begin{pmatrix} \mathcal{I}_u & 0 \\ \mathcal{E}S & \varepsilon \mathcal{I}_W \end{pmatrix}, \quad \mathcal{T}^* = \begin{pmatrix} \mathcal{I}_u & \mathcal{S}^* \mathcal{E}^* \\ 0 & \varepsilon \mathcal{I}_W \end{pmatrix}.
\]

This simplification has to be taken in account dealing with operator \( \mathcal{T} \) and \( \mathcal{T}^* \). \( \diamond \)
1. The linear-quadratic optimal control problem

Remark 1.23. The operator $S^* : W_0(0, T)' \to \mathcal{U}$ denote the dual operator of $S$ and it satisfies

$$\langle S^*g, u \rangle_u = \langle g, Su \rangle_{W(0, T)', W(0, T)}$$

for all $(g, u) \in W(0, T)' \times \mathcal{U}$.

Then, it follows from [127, Lemma 2.4] that for $u \in \mathcal{U}_{ad}$ the state $y = \hat{y} + Su$ and the adjoint $p = \hat{p} + A_1 u$ satisfy

$$B^*(\hat{p}(t)) = (S^*(\Xi(y_Q, y_T) - \Theta(y))) \in \mathbb{R}^m,$$

where $\Xi : L^2(0, T; H) \times H \to W(0, T)'$ and $\Theta : W(0, T) \to W(0, T)'$ are given by

$$\langle \Xi(\chi_1, \chi_2), \varphi \rangle_{W(0, T)', W(0, T)} = \sigma_Q \langle \chi_1, \varphi \rangle_{L^2(0, T; H)} + \sigma_T \langle \chi_2, \varphi(T) \rangle_H,$$

$$\langle \Theta(\chi), \varphi \rangle_{W(0, T)', W(0, T)} = \sigma_Q \langle \chi, \varphi \rangle_{L^2(0, T; H)} + \sigma_T \langle \chi(T), \varphi(T) \rangle_H$$

for $\chi, \varphi \in W(0, T)$ and $(\chi_1, \chi_2) \in L^2(0, T; H) \times H$. In particular, we have $B^*(\hat{p}(t)) = (S^*\Xi(\hat{y}_Q, \hat{y}_T))(t)$. \hfill \Box

With $z_a = (u_a, \hat{y}_a)$, $z_b = (u_b, \hat{y}_b) \in \mathcal{Z}$ we define the closed, convex set of admissible controls as

$$\mathcal{Z}_{ad} = \{ z = (u, w) \in \mathcal{Z} \mid z_a \leq z \leq z_b, w_a \leq w \leq w_b \}.$$ 

Moreover, we introduce the reduced cost functional

$$\hat{J}(z) = J(\hat{y} + Su, u, w)$$

$$= \frac{\sigma_Q}{2} \int_0^T \| (Su)(t) - \hat{y}_Q(t) \|_H^2 \, dt + \frac{\sigma_T}{2} \| (Su)(T) - \hat{y}_T \|_H^2$$

$$+ \frac{1}{2} \sum_{i=1}^m \sigma_i \| u_i \|_{L^2(0, T)}^2 + \frac{\sigma_w}{2} \| w \|_{W}^2.$$ 

Then, $(P)$ is equivalent to the following reduced problem:

$$\min \hat{J}(z) \quad \text{s.t.} \quad z \in \mathcal{Z}_{ad}. \quad (\hat{P})$$

Lemma 1.24. Let Assumptions 1.1, 1.12 and 1.14 be satisfied. Then, for all $\varepsilon \geq 0$, $\tilde{z} = (\tilde{u}, 0)$ is feasible for $(\hat{P})$ with $\tilde{u}$ feasible point for $(\hat{P})$.

Proof. Since $\tilde{u}$ is a feasible point for $(\hat{P})$, it satisfies $u_{ai}(t) \leq \tilde{u}_i(t) \leq u_{bi}(t)$ a.e. in $[0, T]$ for $i = 1, \ldots, m$. Moreover, by assumption $w_a < 0 < w_b$. Furthermore,

$$\hat{y}_a + \tau \leq \tilde{S} \tilde{u} = \tilde{S}u + \varepsilon \leq \hat{y}_b - \tau < \hat{y}_b,$$

which proves the claim. \hfill \Box
Theorem 1.25. Let Assumptions 1.1, 1.12 and 1.14 hold. Then, there exists a unique \((\bar{w}^\varepsilon, \bar{w}^\varepsilon)\) optimal solution of \((\bar{P})\) for every \(\varepsilon > 0\).

Proof. The existence of a feasible point is proved in Lemma 1.24. The set \(Z_{ad}\) is also convex and close; cf. [64]. Moreover, \(\sigma_i > 0\) for all \(i = 1, \ldots, m\), therefore we can again apply Theorem 1.43 in [67, pp.53-54] that completes the proof. \(\square\)

1.3.1 First-order optimality conditions

Theorem 1.26. Let Assumptions 1.1, 1.12 and 1.14 hold. Suppose that \(\bar{z}^\varepsilon = (\bar{w}^\varepsilon, \bar{w}^\varepsilon) \in Z_{ad}\) is the solution to \((\bar{P})\) with associated optimal state \(\bar{y}^\varepsilon = \bar{y} + \mathcal{S}\bar{w}^\varepsilon\). Then, there exist unique Lagrange multipliers \(\bar{p}^\varepsilon \in W(0,T)\) and \(\bar{\beta}^\varepsilon, \bar{\alpha}^\varepsilon \in W\), \(\bar{\alpha}^\varepsilon = (\bar{\alpha}_i^\varepsilon)_{1 \leq i \leq m} \in \mathcal{U}\) satisfying the dual equations

\[
\frac{d}{dt} \langle \bar{p}^\varepsilon(t), \varphi \rangle_H + a(t; \varphi, \bar{p}^\varepsilon(t)) = \sigma_Q \langle (y_Q - \bar{y}^\varepsilon)(t), \varphi \rangle_H - \langle \bar{\beta}^\varepsilon(t), \varphi \rangle_H, \quad \text{for all } \varphi \in V \text{ and a.e. in } [0,T] \text{ and the optimality system}
\]

\[
\begin{align*}
\sigma_i \bar{w}_i^\varepsilon - \gamma_c \int_{\Gamma_c} b_i \bar{p}^\varepsilon \, ds + \bar{\alpha}_i^\varepsilon &= 0 \quad \text{in } L^2(0,T) \text{ for } i = 1, \ldots, m, \\
\sigma_w \bar{w}^\varepsilon + \varepsilon \bar{\beta}^\varepsilon + \bar{\alpha}^\varepsilon &= 0 \quad \text{in } W.
\end{align*}
\]

Moreover,

\[
\begin{align*}
\bar{\beta}_i^\varepsilon &= \max \left\{0, \bar{\beta}_i^\varepsilon + \eta(\bar{y}^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_a)\right\} + \min \left\{0, \bar{\beta}_i^\varepsilon + \eta(y_a^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_a)\right\}, \\
\bar{\alpha}_i^\varepsilon &= \max \left\{0, \bar{\alpha}_i^\varepsilon + \eta_i(\bar{w}_i^\varepsilon - u_{bi})\right\} + \min \left\{0, \bar{\alpha}_i^\varepsilon + \eta_i(\bar{w}_i^\varepsilon - u_{bi})\right\}, \\
\bar{\alpha}_i^\varepsilon &= \max \left\{0, \bar{\alpha}_i^\varepsilon + \eta_w(\bar{w}^\varepsilon - w_b)\right\} + \min \left\{0, \bar{\alpha}_i^\varepsilon + \eta_w(\bar{w}^\varepsilon - w_a)\right\}
\end{align*}
\]

for \(i = 1, \ldots, m\) and for arbitrarily chosen \(\eta, \eta_w, \eta_1, \ldots, \eta_m > 0\), where the max- and min-operations are interpreted componentwise in the pointwise everywhere sense.

Proof. The proof of this result can be obtained following, e.g., [67, 126]. \(\square\)

Remark 1.27. 1) Note that (1.31c) is a NCP function based reformulation of the complementarity system

\[
\begin{align*}
\bar{\beta}_a^\varepsilon &\geq 0, & y_a - \bar{y}^\varepsilon - \varepsilon \bar{w}^\varepsilon &\leq 0, & \langle \bar{\beta}_a^\varepsilon, y_a - \bar{y}^\varepsilon - \varepsilon \bar{w}^\varepsilon \rangle_W &= 0, \\
\bar{\beta}_b^\varepsilon &\geq 0, & \bar{y}^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_b &\leq 0, & \langle \bar{\beta}_b^\varepsilon, \bar{y}^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_b \rangle_W &= 0
\end{align*}
\]
with \( \tilde{\beta}^\varepsilon = \tilde{\beta}_b^\varepsilon - \tilde{\beta}_a^\varepsilon \in W \). Analogously, (1.31d) is a NCP function based reformulation as stated also in Remark 1.19. Furthermore, (1.31e) is a NCP function based reformulation of the complementarity system

\[
\begin{align*}
\bar{\vartheta}_b^\varepsilon \geq 0, & \quad w_a - \bar{w}^\varepsilon \leq 0, \quad \langle \bar{\vartheta}_a^\varepsilon, w_a - \bar{w}^\varepsilon \rangle_W = 0, \\
\bar{\vartheta}_b^\varepsilon \geq 0, & \quad \bar{w}^\varepsilon - w_b \leq 0, \quad \langle \bar{\vartheta}_a^\varepsilon, \bar{w}^\varepsilon - w_b \rangle_W = 0
\end{align*}
\]

with \( \bar{\vartheta}^\varepsilon = \bar{\vartheta}_b^\varepsilon - \bar{\vartheta}_a^\varepsilon \in W \).

2) For given \( \beta \in W \) we define \( p = A_2 \beta \in W(0,T) \), which uniquely solves

\[
- \frac{d}{dt} \langle p(t), \varphi \rangle_H + a(\varphi, p(t)) = - \langle \beta(t), \varphi \rangle_H \text{ for all } \varphi \in V \text{ a.e. in } [0,T],
\]

\[
p(T) = - \beta(T) \quad \text{in } H.
\]

In particular, the solution \( \bar{p}^\varepsilon \) to (1.30) is given by \( \bar{p}^\varepsilon = \hat{p} + A_1 \bar{u}^\varepsilon + A_2 \tilde{\beta}^\varepsilon \in W(0,T) \), where the quantities \( \hat{p} \) and \( A_1 \) are defined in Remark 1.20. \( \diamond \)

It follows from Theorem 1.26 that the first-order conditions for \( \hat{P} \) can be equivalently written as the nonsmooth nonlinear system

\[
\begin{align*}
\sigma_i \bar{u}_i^\varepsilon - \gamma_i \int_{\Gamma_x} b_i \bar{r}^\varepsilon \, ds + \bar{\alpha}_i^\varepsilon &= 0, \quad i = 1, \ldots, m, \quad (1.32a) \\
\sigma_w \bar{w}^\varepsilon + \varepsilon \bar{\beta}^\varepsilon + \bar{\vartheta}^\varepsilon &= 0, \quad (1.32b) \\
\bar{\alpha}_i^\varepsilon &= \max \left\{ 0, \bar{\alpha}_i^\varepsilon + \eta_i (\bar{u}_i^\varepsilon - u_a) \right\} + \min \left\{ 0, \bar{\alpha}_i^\varepsilon + \eta_i (\bar{u}_i^\varepsilon - u_a) \right\}, \quad (1.32c) \\
\bar{\beta}^\varepsilon &= \max \left\{ 0, \bar{\beta}^\varepsilon + \eta (\bar{y}^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_b) \right\} + \min \left\{ 0, \bar{\beta}^\varepsilon + \eta (\bar{y}^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_b) \right\}, \quad (1.32d) \\
\bar{\vartheta}^\varepsilon &= \max \left\{ 0, \bar{\vartheta}^\varepsilon + \eta_w (\bar{w}^\varepsilon - w_b) \right\} + \min \left\{ 0, \bar{\vartheta}^\varepsilon + \eta_w (\bar{w}^\varepsilon - w_b) \right\}, \quad (1.32e)
\end{align*}
\]

with the unknowns \( \bar{u}^\varepsilon, \bar{w}^\varepsilon, \bar{\alpha}^\varepsilon, \bar{\beta}^\varepsilon \) and \( \bar{\vartheta}^\varepsilon \) and the dependent variables \( \bar{y}^\varepsilon = \hat{y} + S \bar{u}^\varepsilon \), \( \bar{p}^\varepsilon = \hat{p} + A_1 \bar{u}^\varepsilon + A_2 \tilde{\beta}^\varepsilon \).

### 1.3.2 Gradient of the reduced cost functional

Suppose that we choose an arbitrary \( z = (u, w) \in \mathbb{Z}_{ad} \). Then we set \( y = \hat{y} + S u \) and \( p = \hat{p} + A_1 u \). It follows by standard arguments that the reduced cost functional is Fréchet-differentiable. In particular, we have for any direction \( z^\delta = (u^\delta, w^\delta) \in \mathbb{Z} \)

\[
\begin{align*}
\tilde{J}(z) z^\delta &= \tilde{J}_u(z) u^\delta + \tilde{J}_w(z) w^\delta \\
&= J_y(y, u, w) S u^\delta + J_u(y, u, w) u^\delta + J_w(y, u, w) w^\delta.
\end{align*}
\]
Note that
\[
J_u(y, u, w)u^\delta = \int_0^T \sum_{i=1}^m \sigma_i u_i(t) u_i^\delta(t) \, dt,
\]
\[
J_w(y, u, w)w^\delta = \int_0^T \sigma_w \langle w(t), w^\delta(t) \rangle_H \, dt.
\] (1.34)

Moreover, setting \( y^\delta = Su^\delta \in W_0(0, T) \) we derive
\[
J_y(y, u, w)Su^\delta = \int_0^T \sigma_Q \langle y(t) - y_Q(t), y^\delta(t) \rangle_H \, dt + \sigma_T \langle y(T) - y_T, y^\delta(T) \rangle_H.
\] (1.35)

**Lemma 1.28.** Suppose that \( z = (u, w) \in Z_{ad} \), \( y = \hat{y} + Su \) and \( p = \hat{p} + A_1 u \). Moreover, \( y^\delta = Su^\delta \) for \( u^\delta \in U \). Then,
\[
\int_0^T \sigma_Q \langle (y - y_Q)(t), y^\delta(t) \rangle_H \, dt
= \sigma_T \langle y_T - y_T, y^\delta(T) \rangle_H - \int_0^T \sum_{i=1}^m \gamma c u_i^\delta(t) \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt.
\]

**Proof.** This lemma can be proven similarly to Lemma 1.21. \( \square \)

From Lemma 1.28, (1.35) and (1.2) we infer that
\[
J_y(y, u, w)Su^\delta = -\int_0^T \sum_{i=1}^m \gamma c u_i^\delta(t) \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt
= -\langle B^* p, u^\delta \rangle_{L^2(0,T;V'),L^2(0,T;V)}.
\] (1.36)

Combining (1.34) and (1.36) we get for the derivative \( \hat{J}'(z) \in \mathcal{Z}' \) at \( z = (u, w) \in Z_{ad} \) the formula
\[
\hat{J}'(z)z^\delta = \int_0^T \sum_{i=1}^m \left( \sigma_i u_i(t) - \gamma c \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \right) u_i^\delta(t) \, dt
+ \int_0^T \sigma_w \langle w(t), w^\delta(t) \rangle_H \, dt.
\] (1.37)

for every \( z^\delta = (u^\delta, w^\delta) \in \mathcal{W} \) with \( p = \hat{p} + A_1 u \). Moreover, the gradient \( \nabla \hat{J}(z) \in \mathcal{Z} \) – the Riesz representant of \( \hat{J}'(z) \) – reads
\[
(\nabla \hat{J}(z))(\cdot) = \begin{pmatrix} \left( \sigma_i u_i(\cdot) - \gamma c \langle b_i, p(\cdot) \rangle_{L^2(\Gamma_c)} \right)_{1 \leq i \leq m} \\ \sigma_w w(\cdot) \end{pmatrix},
\] (1.38)

with \( p = \hat{p} + A_1 u \).
1.3.3 Regularization Error

The strategy introduced in Section 1.3 belongs to a general setting known as virtual control approach; cf. [83]. Starting from the problem (1.15) one can consider the generalized problem

$$
\min J^\varepsilon(y, u, w) := \sigma Q \int_0^T \|y(t) - y_Q(t)\|^2_H dt + \frac{\sigma_T}{2} \|y(T) - y_T\|^2_H + \frac{1}{2} \sum_{i=1}^m \sigma_i \|u_i\|^2_{L^2(0,T)} + \frac{\kappa_1(\varepsilon)}{2} \|w\|^2_W
$$

subject to the modified state equations

$$
y_i(t, x) - \lambda \Delta y_i(t, x) + v_i(t, x) \cdot \nabla y_i(t, x) = \kappa_2(\varepsilon) w(t, x) \quad \text{a.e. in } Q,
$$

$$
\lambda \frac{\partial y_i}{\partial n}(t, s) + \gamma_c y_i(t, s) = \gamma_c \sum_{i=1}^m u_i(t) b_i(s) \quad \text{a.e. on } \Sigma_c,
$$

$$
\lambda \frac{\partial y_i}{\partial n}(t, s) + \gamma_o y_i(t, s) = \gamma_o y_{out}(t) \quad \text{a.e. on } \Sigma_o,
$$

$$
y(0, x) = y_o(x), \quad \text{a.e. in } \Omega
$$

and to the modified inequality constraints

$$
u_{ai}(t) \leq u_i(t) \leq u_{bi}(t) \quad \text{a.e. in } [0, T] \text{ for } i = 1, \ldots, m, \quad (1.39c)
$$

$$
y_a(t, x) \leq y(t, x) + \kappa_3(\varepsilon) w(t, x) \leq y_b(t, x) \quad \text{a.e. in } Q_T, \quad (1.39d)
$$

$$
w_a \leq w(t, x) \leq w_b \quad \text{a.e. in } Q_T, \quad (1.39e)
$$

where $w_a, w_b \in \mathbb{R}$ and $\kappa_1, \kappa_2$ and $\kappa_3$ are chosen non-negative continuous functions defined for $\varepsilon \geq 0$. As in Section 1.3, we can assume without loss of generality that $w_a < 0$ and $w_b > 0$. In [83] convergence of a solution $\bar{x}^\varepsilon = (\bar{y}^\varepsilon, \bar{u}^\varepsilon, \bar{w}^\varepsilon) \in X$ of (1.39) to a solution $\bar{x} = (\bar{y}, \bar{u}, 0) \in X$ of (1.15) is proved for $\varepsilon \to 0$ in the case of an elliptic state equation and unilateral state constraints. In this section, we extend the arguments in [83] to our parabolic setting and to bilateral state constraints.

**Remark 1.29.** We introduce the time-dependent linear operator $W(t) : V \to V'$

$$
\langle W(t), \varphi \rangle_{V', V} = \kappa_2(\varepsilon) \int_\Omega w(t) \varphi dx.
$$

For fixed $\varepsilon \geq 0$ and $w_a \leq w \leq w_b$, the mapping $W$ belongs to $L^2(0,T; V')$, in fact
for every $\varphi \in L^2(0, T; V)$
\[
\int_0^T \langle W(t), \varphi(t) \rangle_{V', V} \, dt = \int_0^T \kappa_2(\varepsilon) w(t) \varphi(t) \, dx \, dt
\leq \kappa_2(\varepsilon) \int_0^T \|w(t)\|_H \|\varphi(t)\|_H \, dt.
\]
Choosing $c_W = \kappa_2(\varepsilon) \max(|w_a|, w_b) (\int_{\Omega} \, dx)^{1/2}$, we have that
\[
\|W\|_{L^2(0, T; V')} = \sup_{\|\varphi\|_{L^2(0, T; V')} = 1} \int_0^T \langle W(t), \varphi(t) \rangle_{V', V} \, dt \leq c_W,
\]
which proves the claim. The weak formulation of (1.39b) is the following:
\[
\frac{d}{dt}(y(t), \varphi)_H + a(t; y(t), \varphi) = \langle F(t) + W(t) + B(u(t)), \varphi \rangle_{V', V}
\quad \text{for all } \varphi \in V \text{ a.e. in } (0, T], \tag{1.41}
\]
\[
y(0) = y_0 \quad \text{in } H.
\]
If Assumption 1.1 is satisfied, (1.41) admits a unique weak solution $y^\varepsilon = \hat{y} + Su + \tilde{S}(\kappa_2(\varepsilon)w)$, where $\tilde{S} : W \to W_0(0, T)$ is the linear solution operator such that for $q \in W$ the function $y = \tilde{S}q$ solves in a weak sense
\[
y_t(x, t) - \lambda \Delta y(x, t) + v(t, x) \cdot \nabla y(t, x) = q(t, x) \quad \text{a.e. in } Q,
\]
\[
\lambda \frac{\partial y}{\partial n}(t, s) + \gamma e y(t, s) = 0 \quad \text{a.e. on } \Sigma_e,
\]
\[
\lambda \frac{\partial y}{\partial n}(t, s) + \gamma o y(t, s) = 0 \quad \text{a.e. on } \Sigma_o,
\]
\[
y(0, x) = 0 \quad \text{a.e. in } \Omega. \tag{1.42}
\]
Clearly, $y^\varepsilon = \hat{y} + Su + \tilde{S}(\kappa_2(\varepsilon)w)$ solves (1.41) and $y = \hat{y} + Su + \tilde{S}(\kappa_2(\varepsilon)0)$ solves (1.3). Moreover, if $y$ is the solution of (1.41), it satisfies the estimate
\[
\|y\|_{W(0, T)} \leq C_y \|y_0\|_H + \|y_{out}\|_{L^2(0, T)} + \|u\|_U + \kappa_2(\varepsilon) \|w\|_W
\]
for a constant $C_y \geq 0$ which is independent of $y_0$, $y_{out}$, $u$, $\varepsilon$ and $w$. \hfill \Diamond

We can now define the reduced cost functional
\[
\tilde{F}^\varepsilon(u, w) := \tilde{F}^\varepsilon(y^\varepsilon(u, w), u, w)
\]
\[
= \frac{\sigma_Q}{2} \int_0^T \|\tilde{S}u + \kappa_2(\varepsilon)\tilde{S}w(t) - \hat{y}_Q(t)\|^2_H \, dt + \frac{1}{2} \sum_{i=1}^m \sigma_i \|u_i\|^2_{L^2(0, T)}
\]
\[
+ \frac{\sigma_T}{2} \|\tilde{S}u + \kappa_2(\varepsilon)\tilde{S}w(T) - \hat{y}_T\|^2_H + \frac{\kappa_1(\varepsilon)}{2} \|w\|^2_W
\]
and the admissible set
\[ Z_{ad}^\varepsilon = \{ z = (u, w) \in U \times W | u_i(t) \leq u_i(t) \leq u_{bi}(t), i = 1, \ldots, m \text{ and a.e. in } [0, T], \]
\[ \hat{y}_a \leq Su + \kappa_2(\varepsilon)\tilde{S}w + \kappa_3(\varepsilon)w \leq \hat{y}_b \text{ a.e. in } Q_T, w_a \leq w \leq w_b \text{ a.e. in } Q_T \}, \]
respectively. Thus, (1.39) is equivalent to
\[ \min \hat{J}(z) \text{ s.t. } z \in Z_{ad}^\varepsilon \]
From now on we follow [83], in order to apply their results to (\( \hat{P}^\varepsilon \)).

**Lemma 1.30.** Let Assumption 1.14 be satisfied. Then, for all \( \varepsilon \geq 0 \), \( \tilde{z} = (\tilde{u}, 0) \) is feasible for (\( \hat{P}^\varepsilon \)) with \( \tilde{u} \) feasible point for (\( \hat{P} \)).

**Proof.** To get the claim, we can proceed as in the proof of Lemma 1.24. \( \square \)

Applying again standard arguments [67, 92], one can prove that there exist a unique optimal solution \( \tilde{z}^\varepsilon = (\tilde{u}^\varepsilon, \tilde{w}^\varepsilon) \in Z_{ad}^\varepsilon \) to (\( \hat{P}^\varepsilon \)). The uniqueness follows from the strict convexity property of the reduced cost functional \( \hat{J}(z) \) on \( Z_{ad}^\varepsilon \).

**Remark 1.31.** If we choose \( \kappa_1(\varepsilon) = \sigma_w, \kappa_2(\varepsilon) = 0 \) and \( \kappa_3(\varepsilon) = \varepsilon \), for fixed \( \varepsilon > 0 \) problem (\( \hat{P}^\varepsilon \)) and problem (\( \hat{P} \)) coincide. Therefore, as already said, problem (\( \hat{P} \)) belongs to the general class of virtual control approach problems. We can follow the proof in [83] to prove convergence of our method in the parabolic setting. \( \diamond \)

To proceed, we want to have an estimation for the term \( \tilde{S}\tilde{w}^\varepsilon \). For this purpose, we apply Lemma 1.8 to (1.42) with \( q = \kappa_2(\varepsilon)w \). In fact, since for fixed \( \varepsilon > 0 \), \( w_{a} \leq \tilde{w}^\varepsilon(t, x) \leq w_{b} \) a.e in \( Q \) we have that \( \tilde{w}^\varepsilon \in L^\infty(Q) \subset L^r(Q) \) for all \( r \in [1, +\infty) \) and therefore in particular it holds for \( r > d/2 + 1 \). Clearly, \( 0 \in L^s(\Sigma), 0 \in L^s(\Sigma) \) for all \( s \geq 1 \) and \( y_0 = 0 \in C(\overline{Q}) \), thus we have that \( y = \kappa_2(\varepsilon)\tilde{S}\tilde{w}^\varepsilon \in W(0, T) \cap C(\overline{Q}) \) and the following estimate holds true for \( r > d/2 + 1 \):
\[ \kappa_2(\varepsilon)(\|\tilde{S}\tilde{w}^\varepsilon\|_{W[0, T]} + \|\tilde{S}\tilde{w}^\varepsilon\|_{C(\overline{Q})}) \leq C\kappa_2(\varepsilon)\|\tilde{w}^\varepsilon\|_{L^r(Q)} \] \[ (1.43) \]

**First-order optimality conditions**

Similarly to Section 1.3.1, we can derive the first-order optimality conditions and the dual equation for (\( \hat{P}^\varepsilon \)). They are resumed in the following theorem:
Theorem 1.32. Let Assumptions 1.1, 1.12 and 1.14 hold. Let $\bar{z}^\varepsilon = (\bar{u}^\varepsilon, \bar{w}^\varepsilon) \in Z_{2d}^\varepsilon$ be the solution to $(\tilde{P}^\varepsilon)$ with associated optimal state $\tilde{y}^\varepsilon = \tilde{y}^\varepsilon = \tilde{y} + S\bar{u}^\varepsilon + \tilde{S}\bar{w}^\varepsilon$. Then, there exist unique Lagrange multipliers $\bar{p}^\varepsilon \in W(0, T)$ and $\bar{\beta}^\varepsilon, \bar{\vartheta}^\varepsilon \in W$, $\bar{\alpha}_i^\varepsilon = (\bar{\alpha}_i^\varepsilon)_{1 \leq i \leq m} \in \mathcal{U}$ satisfying the dual equations

$$
-\frac{d}{dt} \langle \bar{p}^\varepsilon(t), \varphi \rangle_H + a(t; \varphi, \bar{p}^\varepsilon(t)) = \sigma_Q \langle y_Q(t) - \tilde{y}^\varepsilon(t), \varphi \rangle_H - \langle \bar{\beta}^\varepsilon(t), \varphi \rangle_H
$$

$$
p(T) = \sigma_T(y_T - \tilde{y}^\varepsilon(T) - \bar{\beta}^\varepsilon(T)
$$

for all $\varphi \in V$ and a.e. in $[0, T]$ and the optimality system

$$
\sigma_i \bar{u}_i^\varepsilon - \gamma_c \int_{\Gamma_e} b_i \bar{p}^\varepsilon \, ds + \bar{\alpha}_i^\varepsilon = 0 \quad \text{in } L^2(0, T) \text{ for } i = 1, \ldots, m,
$$

$$
\kappa_1(\varepsilon) \bar{w}^\varepsilon + \kappa_3(\varepsilon) \bar{\beta}^\varepsilon + \bar{\vartheta}^\varepsilon = \kappa_2(\varepsilon) \bar{p}^\varepsilon \quad \text{in } \mathcal{W}.
$$

Moreover,

$$
\bar{\beta}^\varepsilon = \max \left\{ 0, \bar{\beta}^\varepsilon + \eta(y^\varepsilon + \kappa_3(\varepsilon) \bar{w}^\varepsilon - y_b) \right\}
$$

$$
+ \min \left\{ 0, \bar{\beta}^\varepsilon + \eta(y^\varepsilon + \kappa_3(\varepsilon) \bar{w}^\varepsilon - y_a) \right\},
$$

$$
\bar{\alpha}_i^\varepsilon = \max \left\{ 0, \bar{\alpha}_i^\varepsilon + \eta_i(\bar{u}_i^\varepsilon - u_{bi}) \right\} + \min \left\{ 0, \bar{\alpha}_i^\varepsilon + \eta_i(\bar{u}_i^\varepsilon - u_{ai}) \right\}
$$

$$
\bar{\vartheta}^\varepsilon = \max \left\{ 0, \bar{\vartheta}^\varepsilon + \eta_w(\bar{w}^\varepsilon - w_b) \right\} + \min \left\{ 0, \bar{\vartheta}^\varepsilon + \eta_w(\bar{w}^\varepsilon - w_a) \right\}
$$

for $i = 1, \ldots, m$ and for arbitrarily chosen $\eta, \eta_w, \eta_1, \ldots, \eta_m > 0$.

Proof. The proof of this result can be obtained following, e.g., [67, 126].

Remark 1.33. We define the linear and bounded operator $\mathcal{A}_3 : \mathcal{W} \rightarrow W(0, T)$ as follows: $p = \mathcal{A}_3 w$ is the unique solution to

$$
-\frac{d}{dt} \langle p(t), \varphi \rangle_H + a(t; \varphi, p(t)) = -\sigma_Q \langle (\tilde{S}w)(t), \varphi \rangle_H
$$

$$
p(T) = -\sigma_T(\tilde{S}w)(T)
$$

for all $\varphi \in V$, a.e. in $[0, T]$ and for given $w \in \mathcal{W}$. Thus, the solution to (1.44) can be expressed as $\bar{p}^\varepsilon = \bar{p} + \mathcal{A}_1 \bar{u}^\varepsilon + \mathcal{A}_2 \bar{\beta}^\varepsilon + \mathcal{A}_3 \bar{w}^\varepsilon$.

In what follows, we derive the reduced gradient of $\tilde{J}^\varepsilon$. The procedure is similar to the one applied in Section 1.3.2, but we need to precise small details. We start from problem $(\tilde{P}^\varepsilon)$ and suppose that we choose an arbitrary $z = (u, w) \in Z_{2d}$. Then we set $y = \tilde{y} + S u + \tilde{S}w$ and $p = \tilde{p} + \mathcal{A}_1 u + \mathcal{A}_3 w$. It follows by standard arguments
that the reduced cost functional $\hat{J}^\varepsilon$ is Fréchet-differentiable. In particular, we have for any direction $z^\eta = (u^\eta, w^\eta) \in \mathcal{Z}$

$$
\left( \hat{J}^\varepsilon \right)'(z)z^\eta = \partial_u \hat{J}^\varepsilon(z)u^\eta + \partial_w \hat{J}^\varepsilon(z)w^\eta
= \partial_u \hat{J}(y, u, w)(Su^\eta + \tilde{S}w^\eta) + \partial_u \hat{J}(y, u, w)u^\eta + \partial_w \hat{J}(y, u, w)w^\eta.
$$

(1.47)

Note that

$$
\partial_u \hat{J}^\varepsilon(y, u, w)u^\eta = \int_0^T \sum_{i=1}^m \sigma_i u_i(t)u_i(t) \, dt,
$$

$$
\partial_w \hat{J}^\varepsilon(y, u, w)w^\eta = \kappa_1(\varepsilon) \int_0^T \langle w(t), w^\eta(t) \rangle_H \, dt.
$$

(1.48)

Moreover, setting $y^\eta = Su^\eta + \tilde{S}w^\eta \in W_0(0, T)$ we derive

$$
\partial_y \hat{J}^\varepsilon(y, u, w)(Su^\eta + \tilde{S}w^\eta) = \int_0^T \sigma_Q \langle y(t) - y_Q(t), y^\eta(t) \rangle_H \, dt
+ \sigma_T \langle y(T) - y_T, y^\eta(T) \rangle_H.
$$

(1.49)

**Lemma 1.34.** Suppose that $z = (u, w) \in \mathcal{Z}^\varepsilon$, $y = \hat{y} + Su + \tilde{S}w$ and $p = \hat{p} + A_1u + A_3w$. Moreover, $y^\eta = Su^\eta + \tilde{S}w^\eta$ for $(u^\eta, w^\eta) \in \mathcal{Z}$. Then,

$$
\int_0^T \sigma_Q \langle (y - y_Q)(t), y^\eta(t) \rangle_H \, dt
= - \int_0^T \kappa_2(\varepsilon) \langle w^\eta(t), p(t) \rangle_H + \sum_{i=1}^m \gamma c_i u_i(t) \langle b_i, p(t) \rangle_{L^2(V_c)} \, dt
+ \sigma_T \langle y_T - y(T), y^\eta(T) \rangle_H.
$$

**Proof.** First we notice that $p$ and $y^\eta$ satisfy the following equations:

$$
\int_0^T -\langle p(t), \varphi(t) \rangle_{V', V} + \alpha(t; \varphi(t), p(t)) \, dt
= \int_0^T \sigma_Q \langle (y_Q - y)(t), \varphi(t) \rangle_H \, dt \quad \text{for all } \varphi \in W(0, T),
$$

$$
p(T) = \sigma_T (y_T - y(T)) \quad \text{in } H
$$

(1.50a)
and
\[ \int_0^T \langle y^p(t), \varphi(t) \rangle_{V', V} + a(t; y^p, \varphi(t)) - \kappa_2(\varepsilon) \langle w^p(t), \varphi(t) \rangle_H \, dt = \int_0^T \gamma \sum_{i=1}^m a_i^p(t) \langle b_i, \varphi(t) \rangle_{L^2(\Gamma_c)} \, dt \quad \text{for all } \varphi \in W(0, T), \]
\[ y^p(0) = 0 \quad \text{in } H. \quad (1.51a) \]

Thus, choosing \( \varphi = y^p \) in (1.50a), \( \varphi = p \) in (1.51a), applying integration by parts and using (1.50b) and (1.51b), we find that
\[ \int_0^T \sigma_Q \langle (y - y_Q)(t), y^p(t) \rangle_H \, dt = \int_0^T \langle p(t), y^p(t) \rangle_{V', V} - a(t; y^p(t), p(t)) \, dt \]
\[ = \int_0^T -\langle y^p(t), p(t) \rangle_{V', V} - a(t; y^p(t), p(t)) \, dt + \langle p(T), y^p(T) \rangle_H - \langle p(0), y^p(0) \rangle_H \]
\[ = -\int_0^T \kappa_2(\varepsilon) \langle w^p(t), p(t) \rangle_H + \sum_{i=1}^m \gamma \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt \]
\[ + \sigma_T \langle y_T - y(T), y^p(T) \rangle_H \]
\[ = -\int_0^T \kappa_2(\varepsilon) \langle w^p(t), p(t) \rangle_H + \sum_{i=1}^m \gamma \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt \]
\[ + \sigma_T \langle y_T - y(T), y^p(T) \rangle_H \]
which is the claim. \( \square \)

From Lemma 1.34, (1.49) and (1.2) we infer that
\[ \partial_y \mathcal{J}^\varepsilon(y, u, w)(Su^w + \tilde{S}w^w) = -\langle \kappa_2(\varepsilon) p, w^u \rangle_W - \langle \mathcal{B}^* p, u^w \rangle_W. \quad (1.52) \]

Combining (1.48) and (1.52) we get for the derivative \( \left( \hat{\mathcal{J}}^\varepsilon \right)'(z) \in Z' \) at \( z = (u, w) \in Z_{ad} \) the formula
\[ \left( \hat{\mathcal{J}}^\varepsilon \right)'(z) z^\eta = \int_0^T \sum_{i=1}^m \left( \sigma_i u_i(t) - \gamma \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \right) a_i^p(t) \, dt \]
\[ + \int_0^T \langle \kappa_1(\varepsilon) w(t) - \kappa_2(\varepsilon) p(t), w^p(t) \rangle_H \, dt. \quad (1.53) \]
for every \( z^0 = (w^0, w^0) \in Z \) with \( p = \hat{p} + A_1 u + A_3 w \). Moreover, the gradient \( \nabla \hat{J}^\varepsilon (z) \in Z \) – the Riesz representant of \( \hat{J}^\varepsilon (z) \) – reads

\[
\nabla \hat{J}^\varepsilon (z) = \left( \begin{pmatrix} \sigma_i u_i (\cdot) - \gamma (b_i, p(\cdot))_{L^2 (\Gamma, \varepsilon)} \end{pmatrix}_{1 \leq i \leq m} \right),
\]

with \( p = \hat{p} + A_1 u + A_3 w \). Furthermore, following, e.g., [67] and in [126], we have:

**Theorem 1.35 (Variational Inequalities).** \( \bar{u} \in \hat{U}_{ad} \) is an optimal solution for \( \hat{P} \) if and only if the variational inequality

\[
\langle \hat{J}'(\bar{u}), u - \bar{u} \rangle_U \geq 0 \quad \text{for all } u \in \hat{U}_{ad}
\]

is satisfied. Moreover, \( \bar{z}^\varepsilon = (\bar{u}^\varepsilon, \bar{w}^\varepsilon) \in Z_{ad}^\varepsilon \) is an optimal solution to \( \hat{P}^\varepsilon \) if and only if the following variational inequality

\[
\langle (\hat{J}^\varepsilon)'(\bar{z}^\varepsilon), z - \bar{z}^\varepsilon \rangle_Z \geq 0 \quad \text{for all } z \in Z_{ad}^\varepsilon
\]

holds true.

**Remark 1.36.** According to Section 1.2.2, the inequality (1.55) is equivalent to

\[
\langle \begin{pmatrix} \sigma_1 \bar{u}_1 \\ \vdots \\ \sigma_m \bar{u}_m \end{pmatrix} - B^* \bar{p}, u - \bar{u} \rangle_U \geq 0 \quad \text{for all } u \in \hat{U}_{ad}
\]

and \( \bar{p} = \hat{p} + A_1 \bar{u} \). Moreover, according to (1.53), (1.56) is equivalent to

\[
\langle \begin{pmatrix} \sigma_1 \bar{u}_1^\varepsilon \\ \vdots \\ \sigma_m \bar{u}_m^\varepsilon \end{pmatrix} - B^* \bar{p}^\varepsilon, u - \bar{u}^\varepsilon \rangle_U + \langle \kappa_1 (\varepsilon) \bar{w}^\varepsilon - \kappa_2 (\varepsilon) \bar{p}^\varepsilon, w - \bar{w}^\varepsilon \rangle_W \geq 0
\]

for all \( z = (u, w) \in Z_{ad}^\varepsilon \) and \( \bar{p}^\varepsilon = \hat{p} + A_1 \bar{u}^\varepsilon + A_3 \bar{w}^\varepsilon \).

**Convergence and error estimate**

In general, \( \bar{u}^\varepsilon \) may not be feasible for \( \hat{P} \). Then, to derive the main result of this section, we need the next two lemmas:
Lemma 1.37. Let \( \kappa_2(\varepsilon) > 0 \) hold true. For every \( \varepsilon > 0 \) there exists a constant \( C > 0 \) such that the estimate
\[
\|\tilde{S} \bar{w}^\varepsilon\|_{C(\Omega)} \leq C
\]
(1.59)
is satisfied.

Proof. Starting from (1.43), we obtain
\[
\|\tilde{S} \bar{w}^\varepsilon\|_{C(\Omega)} \leq C_1 \|\bar{w}^\varepsilon\|_{L^r(Q)}
\]
with \( r > d/2 + 1 \). We are now interested to estimate \( \|\bar{w}^\varepsilon\|_{L^r(Q)} \):
\[
\|\bar{w}^\varepsilon\|_{L^r(Q)} = \left( \int_0^T \int_\Omega |\bar{w}^\varepsilon(t, x)|^r \, dx \, dt \right)^{1/r} \leq \max(|w_a|, w_b)(T\Lambda(\Omega))^{1/r} = C_2
\]
with \( \Lambda(\Omega) \) the finite Lebesgue-measure of the bounded domain \( \Omega \). Clearly,
\[
\|\tilde{S} \bar{w}^\varepsilon\|_{C(\Omega)} \leq C_1 C_2 = C,
\]
so that the lemma is proved. \( \square \)

Lemma 1.38. Let \( \tilde{u} \) satisfy Assumption 1.14 and let \( (\bar{w}^\varepsilon, \bar{w}^\varepsilon) \in Z_{ad}^\varepsilon \) be the optimal solution of \( (\tilde{P}^\varepsilon) \) for fixed \( \varepsilon > 0 \). Moreover, let Assumptions 1.1 and 1.12 hold. Then, for every \( \varepsilon > 0 \) there exists a \( \delta^\varepsilon \in (0, 1) \), such that \( u_\delta = (u_{\delta i})_{1 \leq i \leq m} \), where \( u_{\delta i} := (1 - \delta)\bar{u}^\varepsilon_i + \delta \tilde{u}_i \), for \( i = 1, \ldots, m \), is feasible for \( (\tilde{P}) \) for every \( \delta \in [\delta^\varepsilon, 1] \).

Proof. Since the control constraints of the problems \( (\tilde{P}) \) and \( (\tilde{P}^\varepsilon) \) are equal, the convex combination \( u_\delta \) of \( \bar{u}^\varepsilon \) and \( \tilde{u} \) also satisfies the constraints
\[
u_{\delta i}(t) \leq u_{\delta i}(t) \leq u_{b i}(t) \quad \text{a.e. in } [0, T] \text{ and for } i = 1, \ldots, m.
\]
In order to make \( u_\delta \) feasible for \( (\tilde{P}) \), we assume that \( w_\delta \equiv 0 \). Then, according to Remark 1.29, we have that
\[
y_{\delta} = S u_{\delta} = (1 - \delta)\bar{y}^\varepsilon + \delta \tilde{y} - (1 - \delta)\kappa_2(\varepsilon)\tilde{S} \bar{w}^\varepsilon,
\]
with \( \tilde{y} := S \tilde{u} \). Since we want to prove that \( \tilde{y}_a \leq y_\delta \leq \tilde{y}_b \) for all \( \delta \in [\delta^\varepsilon, 1] \), we recall that \( \tilde{y}_a \leq \tilde{y}^\varepsilon + \kappa_3(\varepsilon)\bar{w}^\varepsilon \leq \tilde{y}_b \) and \( \tilde{y}_a + \tau \leq \tilde{y} \leq \tilde{y}_b - \tau \). At first, we prove the left inequality:
\[
y_\delta = (1 - \delta)\bar{y}^\varepsilon + \delta \tilde{y} - (1 - \delta)\kappa_2(\varepsilon)\tilde{S} \bar{w}^\varepsilon
\geq (1 - \delta)(\tilde{y}_a - \kappa_3(\varepsilon)\bar{w}^\varepsilon) + \delta \tilde{y}_a + \delta \tau - (1 - \delta)\kappa_2(\varepsilon)\tilde{S} \bar{w}^\varepsilon
= \tilde{y}_a + \delta \tau - (1 - \delta)(\kappa_2(\varepsilon)\tilde{S} \bar{w}^\varepsilon + \kappa_3(\varepsilon)\bar{w}^\varepsilon).
\]
Since \( w_a \leq \bar{w}^\varepsilon(t, \mathbf{x}) \leq w_b \) a.e. in \( Q_T \), we have \( \kappa_3(\varepsilon)\bar{w}^\varepsilon \leq \kappa_3(\varepsilon)w_b \). Using (1.59) we obtain
\[
\kappa_2(\varepsilon)\bar{S}\bar{w}^\varepsilon + \kappa_3(\varepsilon)\bar{w}^\varepsilon \leq \kappa_3(\varepsilon)w_b + \kappa_2(\varepsilon)C.
\]

We define \( \kappa_4(\varepsilon) = \kappa_3(\varepsilon)w_b + \kappa_2(\varepsilon)C \). If \( \delta\tau - (1 - \delta)\kappa_4(\varepsilon) \geq 0 \), we have \( y_\delta \geq \hat{y}_a \).

Recalling that we suppose \( w_b > 0 \), note that \( \kappa_4(\varepsilon) \geq 0 \) for every \( \varepsilon > 0 \). Thus, for now we can choose
\[
\delta^\varepsilon = \frac{\kappa_4(\varepsilon)}{\kappa_4(\varepsilon) + \tau},
\]

We need still to find a \( \delta^\varepsilon \) such that \( y_\delta \leq \hat{y}_b \) for all \( \delta \in [\delta^\varepsilon, 1] \). As before,
\[
y_\delta = (1 - \delta)\tilde{y} + \delta\tilde{y} - (1 - \delta)\kappa_2(\varepsilon)\bar{S}\bar{w}^\varepsilon
\leq (1 - \delta)(\hat{y}_b - \kappa_3(\varepsilon)\bar{w}^\varepsilon) + \delta\hat{y}_b - \delta\tau - (1 - \delta)\kappa_2(\varepsilon)\bar{S}\bar{w}^\varepsilon
= \hat{y}_b - \delta\tau - (1 - \delta)(\kappa_2(\varepsilon)\bar{S}\bar{w}^\varepsilon + \kappa_3(\varepsilon)\bar{w}^\varepsilon).
\]

Using the fact that \( \kappa_3(\varepsilon)\bar{w}^\varepsilon \geq \kappa_3(\varepsilon)w_a \) and (1.59), we get
\[
\kappa_2(\varepsilon)\bar{S}\bar{w}^\varepsilon + \kappa_3(\varepsilon)\bar{w}^\varepsilon \geq \kappa_4(\varepsilon)w_a - \kappa_2(\varepsilon)C.
\]

We define \( \kappa_5(\varepsilon) = \kappa_3(\varepsilon)w_a - \kappa_2(\varepsilon)C \). If \( \delta\tau + (1 - \delta)\kappa_5(\varepsilon) \geq 0 \), we find \( y_\delta \leq \hat{y}_b \).

Recalling that we suppose \( w_a < 0 \), note that \( \kappa_5(\varepsilon) \leq 0 \) for every \( \varepsilon > 0 \). To summarize, we can choose
\[
\delta^\varepsilon = \max \left( \frac{\kappa_4(\varepsilon)}{\kappa_4(\varepsilon) + \tau}, \frac{-\kappa_5(\varepsilon)}{\tau - \kappa_5(\varepsilon)} \right),
\]

and \((u_\delta, y_\delta)\) is feasible for \((\tilde{P})\) for all \( \delta \in [\delta^\varepsilon, 1] \).

We are ready to prove the main result of this section:

**Theorem 1.39.** Let \((\bar{y}, \bar{u}) \in W(0, T) \times U \) and \((\tilde{y}^\varepsilon, \tilde{u}^\varepsilon, \tilde{w}^\varepsilon) \in X \) be the optimal solution of \((\tilde{P})\) and \((\tilde{P}^\varepsilon)\), respectively. Then, there exist positive constants \( C_1 \) and \( C_2 \) independent of \( \varepsilon \) such that
\[
\sigma_Q \int_0^T (\bar{y}(t) - \tilde{y}^\varepsilon(t))^2 dt + \sigma_T \|\bar{y}(T) - \tilde{y}^\varepsilon(T)\|^2_H +
\quad + \sum_{i=1}^m \sigma_i \int_0^T |\bar{u}_i(t) - \tilde{u}_i^\varepsilon(t)|^2 dt \leq C_1\delta^\varepsilon + C_2\frac{\kappa_2(\varepsilon)^2}{\kappa_1(\varepsilon)}
\]

with \( \delta^\varepsilon \) defined as in (1.61).
Proof. Due to Lemma 1.30, the pair \((\bar{u}, 0)\) is feasible for \((\hat{P}_\varepsilon)\). Thus, we can choose it as test function for the variational inequality (1.58) and obtain that

\[
\left\langle \begin{pmatrix}
\sigma_1 \bar{u}_1 \\
\vdots \\
\sigma_m \bar{u}_m
\end{pmatrix} - B^* \bar{p}, \bar{u} - \bar{u}^\varepsilon \right\rangle_u + \langle \kappa_1(\varepsilon) \bar{w}^\varepsilon - \kappa_2(\varepsilon) \bar{p}, -\bar{w}^\varepsilon \rangle_w \geq 0
\]  

(1.63)

holds true with \(\bar{p} = \hat{p} + A_1 \bar{u}^\varepsilon + A_3 \bar{w}^\varepsilon\). Moreover, choosing \(\delta^\varepsilon\) and \(u_{\delta}\) as in Lemma 1.38, we have that \(u_{\delta} \in \tilde{U}_{\text{ad}}\) for \(\delta \in [\delta^\varepsilon, 1]\). Consequently, \(u_{\delta}\) can be used as test function in the variational inequality (1.57) to get

\[
0 \leq \left\langle \begin{pmatrix}
\sigma_1 \bar{u}_1 \\
\vdots \\
\sigma_m \bar{u}_m
\end{pmatrix} - B^* \bar{p}, u_{\delta} - \bar{u} \right\rangle_u = \left\langle \begin{pmatrix}
\sigma_1 \bar{u}_1 \\
\vdots \\
\sigma_m \bar{u}_m
\end{pmatrix} - B^* \bar{p}, u_{\delta} - \bar{u}^\varepsilon + \bar{u}^\varepsilon - \bar{u} \right\rangle_u
\]  

(1.64)

for all \(\delta \in [\delta^\varepsilon, 1]\) and \(\bar{p} = \hat{p} + A_1 \bar{u}\). Adding (1.63) and (1.64) and rearranging their terms, we find

\[
\langle B^*(\bar{p} - \bar{p}^\varepsilon), \bar{u} - \bar{u}^\varepsilon \rangle_u + \sum_{i=1}^m \sigma_i \left\langle \bar{u}_i - \bar{u}_i^\varepsilon, \bar{u}_i - \bar{u}_i \right\rangle_{L^2(0,T)} + \\
+ \left\langle \begin{pmatrix}
\sigma_1 \bar{u}_1 \\
\vdots \\
\sigma_m \bar{u}_m
\end{pmatrix} - B^* \bar{p}, u_{\delta} - \bar{u}^\varepsilon \right\rangle_u + \langle \kappa_1(\varepsilon) \bar{w}^\varepsilon - \kappa_2(\varepsilon) \bar{p}, -\bar{w}^\varepsilon \rangle_w \geq 0.
\]  

(1.65)

For the first term of (1.65) we have

\[
\langle B^*(\bar{p} - \bar{p}^\varepsilon), \bar{u} - \bar{u}^\varepsilon \rangle_u = \langle B(\bar{u} - \bar{u}^\varepsilon), \bar{p} - \bar{p}^\varepsilon \rangle_{L^2(0,T;V')} + \langle \bar{w}^\varepsilon, \bar{p} - \bar{p}^\varepsilon \rangle_{H}
\]  

\[
= \int_0^T \langle B(\bar{u}(t) - \bar{u}^\varepsilon(t)), \bar{p}(t) - \bar{p}^\varepsilon(t) \rangle_{V'\times V} \, dt \\
= \int_0^T \frac{d}{dt} \langle \bar{y}(t) - \bar{y}^\varepsilon(t), \bar{p}(t) - \bar{p}^\varepsilon(t) \rangle_H \, dt \\
+ \int_0^T a(t; \bar{y}(t) - \bar{y}^\varepsilon(t), \bar{p}(t) - \bar{p}^\varepsilon(t)) \, dt \\
+ \int_0^T \kappa_2(\varepsilon) \langle \bar{w}^\varepsilon(t), \bar{p}(t) - \bar{p}^\varepsilon(t) \rangle_H \, dt
\]  

(1.66)

according to (1.3) and (1.41). Using the fact that \(\bar{p}^\varepsilon(T) = \sigma_T(y_T - \bar{y}^\varepsilon(T))\) and
\( \tilde{p}(T) = \sigma_T(y_T - \tilde{y}(T)) \) together with integration by parts, we obtain
\[
\int_0^T \frac{d}{dt} \langle \tilde{y}(t) - \tilde{y}^\varepsilon(t), \tilde{p}(t) - \tilde{p}^\varepsilon(t) \rangle_H dt = \int_0^T \langle \partial_t(\tilde{y}(t) - \tilde{y}^\varepsilon(t)), \tilde{p}(t) - \tilde{p}^\varepsilon(t) \rangle_H dt
\]
\[
= \int_0^T -\langle \partial_t(\tilde{p}(t) - \tilde{p}^\varepsilon(t)), \tilde{y}(t) - \tilde{y}^\varepsilon(t) \rangle_H dt + \langle \tilde{y}(T) - \tilde{y}^\varepsilon(T), \tilde{p}(T) - \tilde{p}^\varepsilon(T) \rangle_H
\]
Then, considering the previous equality and choosing \( \varphi = \tilde{y}(t) - \tilde{y}^\varepsilon(t) \) in the respective definitions of \( \tilde{p}, \mathcal{A}_1 \) and \( \mathcal{A}_3 \), i.e., in Remarks 1.20 and 1.33, we have
\[
\int_0^T \frac{d}{dt} \langle \tilde{y}(t) - \tilde{y}^\varepsilon(t), \tilde{p}(t) - \tilde{p}^\varepsilon(t) \rangle_H dt + a(t; \tilde{y}(t) - \tilde{y}^\varepsilon(t), \tilde{p}(t) - \tilde{p}^\varepsilon(t)) dt
\]
\[
= \int_0^T \frac{d}{dt} \langle \tilde{p}(t) - \tilde{p}^\varepsilon(t), \tilde{y}(t) - \tilde{y}^\varepsilon(t) \rangle_H dt + a(t; \tilde{y}(t) - \tilde{y}^\varepsilon(t), \tilde{p}(t) - \tilde{p}^\varepsilon(t)) dt
\]
\[
+ \sigma_T \langle \tilde{y}(T) - \tilde{y}^\varepsilon(T), \tilde{y}^\varepsilon(T) - \tilde{y}(T) \rangle_H
\]
\[
= \int_0^T \sigma_Q \langle \tilde{y}^\varepsilon(t) - \tilde{y}(t), \tilde{y}(t) - \tilde{y}^\varepsilon(t) \rangle_H dt + \sigma_T \langle \tilde{y}(T) - \tilde{y}^\varepsilon(T), \tilde{y}^\varepsilon(T) - \tilde{y}(T) \rangle_H
\]
\[
= -\sigma_Q \int_0^T \|\tilde{y}(t) - \tilde{y}^\varepsilon(t)\|^2_H dt - \sigma_T \|\tilde{y}(T) - \tilde{y}^\varepsilon(T)\|^2_H.
\]
Thus, using the previous equality in (1.66), we obtain
\[
\langle \mathcal{B}^\ast(\tilde{p} - \tilde{p}^\varepsilon), \tilde{u} - \tilde{u}^\varepsilon \rangle_u = \kappa_2(\varepsilon) \langle \tilde{p} - \tilde{p}^\varepsilon, \tilde{w}^\varepsilon \rangle_w - \sigma_T \|\tilde{y}(T) - \tilde{y}^\varepsilon(T)\|^2_H
\]
\[
- \sigma_Q \int_0^T \|\tilde{y}(t) - \tilde{y}^\varepsilon(t)\|^2_H dt. \quad (1.67)
\]
Inserting (1.67) into (1.65), we get
\[
\sigma_Q \int_0^T \|\tilde{y}(t) - \tilde{y}^\varepsilon(t)\|^2_H dt + \sigma_T \|\tilde{y}(T) - \tilde{y}^\varepsilon(T)\|^2_H + \sum_{i=1}^m \sigma_i \|\tilde{u}_i - \tilde{u}_i^\varepsilon\|^2_{L^2(0,T)}
\]
\[
\leq \left\langle \begin{pmatrix} \sigma_1 \tilde{u}_1 \\ \vdots \\ \sigma_m \tilde{u}_m \end{pmatrix}, -\mathcal{B}^\ast \tilde{p}, \tilde{u} - \tilde{u}^\varepsilon \right\rangle_u - \kappa_1(\varepsilon) \|\tilde{w}^\varepsilon\|^2_w + \kappa_2(\varepsilon) \langle \tilde{p}, \tilde{w}^\varepsilon \rangle_w. \quad (1.68)
\]
Now we are interested to estimate the terms in the right-hand side of inequality (1.68). For the first one, we exploit the definition of \( u_\delta \) in Lemma 1.38. Therefore,

\[
\left\langle \begin{pmatrix} \sigma_1 \bar{u}_1 \\ \vdots \\ \sigma_m \bar{u}_m \end{pmatrix} - B^* \bar{p}, u_\delta - \bar{u}^\varepsilon \right\rangle \leq \sum_{i=1}^m \langle \sigma_i \bar{u}_i(t) - \gamma_\varepsilon \langle b_i, \bar{p}(t) \rangle_{L^2(\Gamma)}, u_{bi} - \bar{u}^\varepsilon_i \rangle_{L^2(0,T)} \leq \sum_{i=1}^m \| \sigma_i \bar{u}_i(t) - \gamma_\varepsilon \langle b_i, \bar{p}(t) \rangle_{L^2(\Gamma)} \|_{L^2(0,T)} \| u_{bi} - \bar{u}^\varepsilon_i \|_{L^2(0,T)}
\]

\[
= \sum_{i=1}^m \bar{C}_i \| \delta(\bar{u}_i - \bar{u}^\varepsilon_i) \|_{L^2(0,T)}
\]

\[
\leq \sum_{i=1}^m \bar{C}_i \| u_{bi} - u_{ai} \|_{L^2(0,T)},
\]

where we have used the Cauchy-Schwarz inequality, the fact that \( \bar{u} \) and \( \bar{u}^\varepsilon \) satisfy the control constraints and \( \bar{C}_i = \| \sigma_i \bar{u}_i(t) - \gamma_\varepsilon \langle b_i, \bar{p}(t) \rangle_{L^2(\Gamma)} \|_{L^2(0,T)} \). Since (1.68) holds for every \( \delta \in [\delta^*, 1] \), we can choose in particular \( \delta = \delta^* \), obtaining

\[
\sigma_\varepsilon \int_0^T \| \bar{g}(t) - \bar{g}^\varepsilon(t) \|_H^2 \, dt + \sigma_T \| \bar{g}(T) - \bar{g}^\varepsilon(T) \|_H^2 + \sum_{i=1}^m \| \bar{u}_i - \bar{u}_i^\varepsilon \|_{L^2(0,T)}^2 \leq \sum_{i=1}^m \bar{C}_i \delta^\varepsilon \| u_{bi} - u_{ai} \|_{L^2(0,T)} - \kappa_1(\varepsilon) \| \bar{w}^\varepsilon \|_W^2 + \kappa_2(\varepsilon) \langle \bar{p}, \bar{w}^\varepsilon \rangle_W.
\]

(1.69)

If \( \kappa_1(\varepsilon), \kappa_2(\varepsilon) > 0 \), we can apply the Cauchy-Schwarz and Young’s inequalities to get

\[
\langle \bar{p}, \bar{w}^\varepsilon \rangle_W \leq \| \bar{p}, \bar{w}^\varepsilon \|_W \leq \frac{\kappa_2(\varepsilon)}{4\kappa_1(\varepsilon)} \| \bar{p} \|_W^2 + \frac{\kappa_1(\varepsilon)}{\kappa_2(\varepsilon)} \| \bar{w}^\varepsilon \|_W^2.
\]

(1.70)

Setting

\[
C_1 = \sum_{i=1}^m \bar{C}_i \| u_{bi} - u_{ai} \|_{L^2(0,T)}, \quad C_2 = \frac{1}{4} \int_0^T \| \bar{p}(t) \|_H^2 \, dt,
\]

we obtain the claim from (1.69) and (1.70). If \( \kappa_1(\varepsilon) > 0 \) and \( \kappa_2(\varepsilon) = 0 \) hold\(^1\), we

\(^1\)We do not discuss the case \( \kappa_1(\varepsilon) = 0 \), since this implies that there is no term depending on \( w \) in \( \mathcal{F}^\varepsilon \). Therefore, we can not ensure the uniqueness of \( w \), when we solve the state constrained problem minimizing \( \mathcal{F}^\varepsilon \).
can follow step by step the previous proof to obtain
\[
\sigma_Q \int_0^T \| \bar{y}(t) - \bar{y}^\varepsilon(t) \|^2_H dt + \sigma_T \| \bar{y}(T) - \bar{y}^\varepsilon(T) \|^2_H + \sum_{i=1}^m \sigma_i \| \bar{u}_i - \bar{u}^\varepsilon_i \|^2_{L^2(0,T)} \\
\leq C_1 \delta^\varepsilon - \kappa_1(\varepsilon) \| \bar{w}^\varepsilon \|^2_W. 
\] (1.71)

Considering the fact that \(-\kappa_1(\varepsilon) \| \bar{w}^\varepsilon \|^2_W \leq 0\) in (1.71), we get anyway the claim. It is also possible to limit the previous constants \(C_1\) and \(C_2\) by expressions which contain only data of the problem, as also mentioned in [83], but we do not go further proving it.

As stated in [83], in practical applications the infeasibility of the regularized control \(\bar{u}^\varepsilon\) with respect to \((\hat{\mathcal{P}})\) may cause problems. Thus, following the same idea of [83] we can prove the same rate of convergence in Theorem 1.39 for \(u_\delta\). Before doing this, we need to prove the following lemma:

**Lemma 1.40.** Let Assumption 1.14 hold. Consider \(\varepsilon > 0, \kappa_1(\varepsilon) > 0\) and let \((\bar{y}^\varepsilon, \bar{u}^\varepsilon, \bar{w}^\varepsilon) \in \mathcal{Z}^\varepsilon_{ad}\) be the optimal solution to problem \((\hat{\mathcal{P}}^\varepsilon)\), then the estimation
\[
\| \bar{w}^\varepsilon \|_W \leq \frac{C}{\sqrt{\kappa_1(\varepsilon)}} 
\] (1.72)
holds true for \(C > 0\) constant independent from \(\varepsilon\).

**Proof.** Since \((\bar{y}^\varepsilon, \bar{u}^\varepsilon, \bar{w}^\varepsilon) \in \mathcal{Z}^\varepsilon_{ad}\) is the optimal solution for problem \((\hat{\mathcal{P}}^\varepsilon)\), we have that
\[
\frac{\kappa_1(\varepsilon)}{2} \| \bar{w}^\varepsilon \|^2_W \leq \mathcal{J}^\varepsilon(\bar{y}^\varepsilon, \bar{u}^\varepsilon, \bar{w}^\varepsilon) \leq \mathcal{J}^\varepsilon(\bar{y}, \bar{u}, 0) = \mathcal{J}(\bar{y}, \bar{u})
\]
with \(\bar{u}\) satisfying the Assumption 1.14 and \(\bar{y} = \hat{y} + \mathcal{S}\bar{u}\). From the previous inequality, we can derive the claim with \(C = \sqrt{2\mathcal{J}(\bar{y}, \bar{u})}\). \(\square\)

Thus, we can prove the next result:

**Corollary 1.41.** Let \(u_\delta\) be the control defined in Lemma 1.38 and let the hypotheses of Lemma 1.38 and Theorem 1.39 be satisfied. Then, for \(\delta = \delta^\varepsilon\) there exist positive constants \(\tilde{C}_1\) and \(\tilde{C}_2\) independent of \(\varepsilon\), such that the estimate
\[
\sigma_Q \int_0^T \| \bar{y}(t) - y_\delta(t) \|^2_H dt + \sigma_T \| \bar{y}(T) - y_\delta(T) \|^2_H + \\
\sum_{i=1}^m \sigma_i \int_0^T \| \bar{u}_i(t) - u_{\delta i}(t) \|^2 dt \leq \tilde{C}_1 \delta^\varepsilon + \tilde{C}_2 \frac{\kappa_2(\varepsilon)^2}{\kappa_1(\varepsilon)} 
\] (1.73)
holds true.
1.3 The virtual control approach

Proof. Using the inequality \((a + b)^2 \leq 2(a^2 + b^2)\), we have

\[
\int_0^T \|\bar{y}(t) - y_\delta(t)\|^2_H dt \leq 2 \int_0^T \|\bar{y}(t) - \bar{y}(t)\|^2_H + \|\bar{y}(t) - y_\delta(t)\|^2_H dt, 
\]

\[
\|\bar{y}(T) - y_\delta(T)\|^2_H \leq 2\|\bar{y}(T) - \bar{y}(T)\|^2_H + 2\|\bar{y}(T) - y_\delta(T)\|^2_H,
\]

\[
\int_0^T |\bar{u}_i(t) - u_{\delta i}(t)|^2 dt \leq 2 \int_0^T |\bar{u}_i(t) - \bar{u}_i(t)|^2 + |\bar{u}_i(t) - \bar{u}_{\delta i}(t)|^2 dt.
\]  

At first, we get

\[
\int_0^T |\bar{u}_i(t) - u_{\delta i}(t)|^2 dt \leq \delta^2 \int_0^T |\bar{u}_i(t) - \bar{u}_i(t)|^2 dt \leq \delta^2 \|u_{bi} - u_{ai}\|^2_{L^2(0,T)},
\]  

because by construction \(u_{\delta i} := (1 - \delta)\bar{u}_i + \delta \tilde{u}_i\) for \(i = 1, \ldots, m\) and by Lemma 1.38 \(u_\delta\) is feasible for \((\tilde{P})\) for every \(\delta \in [\delta^e, 1]\). Hence, it satisfies the control constraints as well as \(\bar{u}\) and \(\tilde{u}\) do. Further, it holds that

\[
\|\bar{y}(t) - y_\delta(t)\|^2_H = \|\delta \bar{y}(t) - \delta \bar{y}(t) + (1 - \delta)\kappa_2(\varepsilon)\tilde{S}\bar{w}(t)\|^2_H
\]

\[
= \|\delta(S\bar{u} + \kappa_2(\varepsilon)\tilde{S}\bar{w}) - (1 - \delta)\kappa_2(\varepsilon)(\tilde{S}\bar{w})\|^2_H
\]

\[
= \|\delta(S(\bar{u} - \bar{u})) - (1 - \delta)\kappa_2(\varepsilon)(\tilde{S}\bar{w})\|^2_H.
\]

a.e. for \(t \in [0, T]\). Applying Proposition 1.5 and Lemma 1.40, we obtain

\[
\int_0^T \|\bar{y}(t) - y_\delta(t)\|^2_H dt \leq \delta^2 C\|\bar{u} - \bar{u}\|^2_H + \tilde{C}\frac{\kappa_2(\varepsilon)^2}{\kappa_1(\varepsilon)}
\]

\[
\leq \delta^2 C\|u_b - u_a\|^2_H + \tilde{C}\frac{\kappa_2(\varepsilon)^2}{\kappa_1(\varepsilon)}
\]

Moreover, using Corollary 1.11 and Lemma 1.40

\[
\|\bar{y}(T) - y_\delta(T)\|^2_H \leq \delta C_T\|\bar{u} - \bar{u}\|^2_H + \tilde{C}T\frac{\kappa_2(\varepsilon)^2}{\kappa_1(\varepsilon)}
\]

\[
\leq \delta^2 C_T\|u_b - u_a\|^2_H + \tilde{C}T\frac{\kappa_2(\varepsilon)^2}{\kappa_1(\varepsilon)}
\]

for \(s > d + 1\) and

\[
U^* := L^s(0, T; \mathbb{R}^m).
\]  

Thus, due to the fact that \(\delta \in [\delta^e, 1]\), we find

\[
\sigma_Q \int_0^T \|\bar{y}(t) - y_\delta(t)\|^2_H dt + \sigma_T\|\bar{y}(T) - y_\delta(T)\|^2_H +
\]

\[
+ \sum_{i=1}^m \sigma_i \int_0^T |\bar{u}_i(t) - u_{\delta i}(t)|^2 dt \leq \tilde{C}_1 \delta^e + \tilde{C}_2 \frac{\kappa_2(\varepsilon)^2}{\kappa_1(\varepsilon)}
\]
with $\tilde{C}_1$ and $\tilde{C}_2$ positive constant independent from $\varepsilon$. 

To have strong convergence in $\mathcal{U}$ of $\bar{u}^\varepsilon$ to $\bar{u}$ as $\varepsilon \to 0$ and of $\bar{y}^\varepsilon$ to $\bar{y}$ in $L^2(0,T;H)$, we have to make the following assumption:

**Assumption 1.42.** The continuous and non-negative functions $\kappa_1(\varepsilon)$, $\kappa_2(\varepsilon)$ and $\kappa_3(\varepsilon)$ satisfy

\[
\lim_{\varepsilon \to 0^+} \kappa_3(\varepsilon) = 0, \quad \lim_{\varepsilon \to 0^+} \frac{\kappa_2(\varepsilon)}{\sqrt{\kappa_1(\varepsilon)}} = 0. \tag{1.77}
\]

Therefore, we have the following result:

**Corollary 1.43.** Let Assumption 1.42 and hypotheses of Theorem 1.39 hold. Moreover, let $\kappa_4(\varepsilon)$, $\kappa_5(\varepsilon)$ and $\delta^\varepsilon$ be defined as in Lemma 1.38. Then, we have the three strong convergences:

\[
\bar{u}^\varepsilon \rightrightarrows \bar{u}, \quad \bar{y}^\varepsilon \overset{L^2(0,T;H)}{\to} \bar{y}, \quad \bar{y}^\varepsilon(T) \overset{H}{\to} \bar{y}(T),
\]

as $\varepsilon \to 0$, where $(\bar{y}, \bar{u}) \in W(0,T) \times \mathcal{U}$ and $(\bar{y}^\varepsilon, \bar{u}^\varepsilon, \bar{w}^\varepsilon) \in \mathcal{X}$ are the optimal solution of (1.15) and (1.39), respectively.

**Proof.** The proof follows directly from Assumption 1.42, Theorem 1.39 and from the fact that $\kappa_4(\varepsilon)$, $\kappa_5(\varepsilon)$, $\delta^\varepsilon \to 0$ as $\varepsilon \to 0$. 

**Remark 1.44.** Due to Corollary 1.41 and Corollary 1.43, the feasible control $u_5^\varepsilon$ converges to $\bar{u}$ and $y_5^\varepsilon$ converges to $\bar{y}$ as $\varepsilon \to 0$. 

**Remark 1.45** (Stronger convergence). Due to Proposition 1.5, Remark 1.29 and Lemma 1.40 we have

\[
\|\bar{y} - \bar{y}^\varepsilon\|_{W(0,T)} = \|S\bar{u} - S\bar{u}^\varepsilon - \kappa_2(\varepsilon)\tilde{S}\bar{w}^\varepsilon\|_{W(0,T)} \leq \hat{C}_1\|\bar{u} - \bar{u}^\varepsilon\|_{\mathcal{U}} + \hat{C}_2\frac{\kappa_2(\varepsilon)}{\sqrt{\kappa_1(\varepsilon)}}
\]

with $\hat{C}_1$ and $\hat{C}_2$ independent from $\varepsilon$. Thus, applying Theorem 1.39 we have that $\bar{y}^\varepsilon \to \bar{y}$ in $W(0,T)$ as $\varepsilon \to 0$.

**Remark 1.46.** In $\hat{\mathcal{P}}$, we have made the choice $\kappa_1(\varepsilon) = \sigma_w$, $\kappa_2(\varepsilon) = 0$ and $\kappa_3(\varepsilon) = \varepsilon$. Since with this choice we satisfy Assumption 1.42, we have convergence
1.3 The virtual control approach

To the solution of the pointwise state constrained optimal control problem. Moreover, using (1.61), Theorem 1.39 reads

\[ \sigma_Q \int_0^T \|\bar{y}(t) - \bar{y}^e(t)\|^2_H dt + \sigma_T \|\bar{y}(T) - \bar{y}^e(T)\|^2_H + \sum_{i=1}^m \sigma_i \int_0^T |\bar{u}_i(t) - \bar{u}_{ie}(t)|^2 dt \leq C \frac{\varepsilon}{\varepsilon + \tau}. \]  

(1.78)

Hence, defining \( \sigma_u = \min\{\sigma_1, \ldots, \sigma_m\} \),

\[ \sigma_u \|\bar{u}^e - \bar{u}\|_U \leq C \frac{\varepsilon}{\varepsilon + \tau} \]

and the expected rate of convergence is \( \|\bar{u}^e - \bar{u}\|_U = O(\sqrt{\varepsilon}) \).

\( \diamond \)

1.3.4 Primal-dual active set strategy

Aware of the convergence of the virtual control approach as \( \varepsilon \to 0 \), to find the solution of the optimality system (1.31) (and therefore the optimal solution of (\( \tilde{P} \))), we utilize the primal-dual active set strategy (PDASS); cf. [63, 64, 74, 75]. In [63], it is proven that the PDASS is equivalent to a semi-smooth Newton method and therefore it is super-linearly convergent. To derive this approach for our problem, we start from the optimality conditions (1.32) and we rearrange them in such a way that the semi-smooth Newton method can be applied. From (1.32a) we have

\[ \bar{\alpha}^e_i = \gamma_c \int_{\Gamma_c} b_i \bar{p}^e \, ds - \sigma_i \bar{u}^e_i \quad \text{a.e. in } [0, T] \text{ for } i = 1, \ldots, m. \]  

(1.79)

Inserting (1.79) into (1.32c) and choosing \( \eta_i = \sigma_i > 0, i = 1, \ldots, m \), we find that

\[ 0 = \gamma_c \int_{\Gamma_c} b_i \bar{p}^e \, ds - \sigma_i \bar{u}^e_i - \max \left\{ 0, \gamma_c \int_{\Gamma_c} b_i \bar{p}^e \, ds - \sigma_i \bar{u}^e_i + \sigma_i (\bar{u}^e_i - u_{bi}) \right\} \]

\[ - \min \left\{ 0, \gamma_c \int_{\Gamma_c} b_i \bar{p}^e \, ds - \sigma_i \bar{u}^e_i + \sigma_i (u_{bi} - \bar{u}^e_i) \right\} \]

\[ = \gamma_c \int_{\Gamma_c} b_i \bar{p}^e \, ds - \sigma_i \bar{u}^e_i - \max \left\{ 0, \gamma_c \int_{\Gamma_c} b_i \bar{p}^e \, ds - \sigma_i u_{bi} \right\} \]

\[ - \min \left\{ 0, \gamma_c \int_{\Gamma_c} b_i \bar{p}^e \, ds - \sigma_i u_{ai} \right\} \quad \text{a.e. in } [0, T] \text{ for } i = 1, \ldots, m. \]

(1.80)

Analogously, (1.32b) yields

\[ \bar{\beta}^e = \frac{\sigma_w}{\varepsilon} \bar{w}^e - \frac{\partial^e}{\varepsilon} \quad \text{a.e. in } Q_T. \]

(1.81)
Setting $\eta = \sigma_w/\varepsilon^2 > 0$ and utilizing (1.32b) we derive from (1.32d)

$$0 = \frac{-\sigma_w}{\varepsilon} \tilde{w}^\varepsilon - \frac{\vartheta^\varepsilon}{\varepsilon} - \max \left\{ 0, \frac{-\sigma_w}{\varepsilon} \tilde{w}^\varepsilon - \frac{\vartheta^\varepsilon}{\varepsilon} + \frac{\sigma_w}{\varepsilon^2} \left( \tilde{y}^\varepsilon + \varepsilon \tilde{w}^\varepsilon - y_b \right) \right\}$$

$$- \min \left\{ 0, \frac{-\sigma_w}{\varepsilon} \tilde{w}^\varepsilon - \frac{\vartheta^\varepsilon}{\varepsilon} + \frac{\sigma_w}{\varepsilon^2} \left( \tilde{y}^\varepsilon + \varepsilon \tilde{w}^\varepsilon - y_b \right) \right\}$$

$$= \frac{-\sigma_w}{\varepsilon} \tilde{w}^\varepsilon - \frac{\vartheta^\varepsilon}{\varepsilon} - \max \left\{ 0, \frac{\sigma_w}{\varepsilon^2} \left( \tilde{y}^\varepsilon - y_b \right) - \frac{\vartheta^\varepsilon}{\varepsilon} \right\}$$

$$- \min \left\{ 0, \frac{\sigma_w}{\varepsilon^2} \left( \tilde{y}^\varepsilon - y_b \right) - \frac{\vartheta^\varepsilon}{\varepsilon} \right\} \text{ a.e. in } Q_T.$$  \hspace{1cm} (1.82)

Unfortunately, the non-smooth equation (1.32e), as also remarked in [74, 75], may cause numerical problem if we apply directly the semi-smooth Newton method. The reason is that the variable $\vartheta^\varepsilon$ appears outside and inside the minimum and maximum operation, therefore it cancels out when $\vartheta^\varepsilon$ is active, implying that the matrix of the semi-smooth Newton method is singular (or close to a singular matrix). Moreover, it may imply jumps in the solution across the boundary of the active and inactive sets, which are not predicted by the theory; cf. [75]. Following [74, 75], we smooth (1.32e) replacing it with

$$0 = \bar{\vartheta}^\varepsilon - \alpha \max \left\{ 0, \bar{\vartheta}^\varepsilon + e_w (\bar{w}^\varepsilon - w_b) \right\} - \alpha \min \left\{ 0, \bar{\vartheta}^\varepsilon + e_w (\bar{w}^\varepsilon - w_a) \right\}$$

for $0 < \alpha < 1$. This is equivalent to

$$0 = \bar{\vartheta}^\varepsilon - \max \left\{ 0, c (\bar{w}^\varepsilon - w_b) \right\} - \min \left\{ 0, c (\bar{w}^\varepsilon - w_a) \right\}$$  \hspace{1cm} (1.83)

setting $c := e_w \alpha / (1 - \alpha)$. Note that (1.83) can be considered as a penalty formulation of the complementarity condition (1.31e) as $c \to +\infty$. At this stage, we have a parameter $\varepsilon$ that has to be sent to zero and a parameter $c$ that has to be sent to infinity in order to have convergence to the solution of ($\hat{P}$). Therefore, assuming without loss of generality that $0 < \varepsilon < 1$, it seems reasonable to make the choice $0 < \alpha = 1 - \varepsilon < 1$ and $e_w = \sigma_w / \alpha > 0$. With these choices $c = \sigma_w / \varepsilon$ and goes to infinity as $\varepsilon \to 0$. Going back to the semi-smooth Newton method, (1.32e) has to be replaced by (1.83) with $c = \sigma_w / \varepsilon$, i.e.

$$0 = \bar{\vartheta}^\varepsilon - \max \left\{ 0, \frac{\sigma_w}{\varepsilon} (\bar{w}^\varepsilon - w_b) \right\} - \min \left\{ 0, \frac{\sigma_w}{\varepsilon} (\bar{w}^\varepsilon - w_a) \right\}.$$  \hspace{1cm} (1.84)

To simplify the notation, the non-smooth operator equations (1.80), (1.82) and (1.84) can be rewritten as a unique one. To do so, for $i = 1, \ldots, m$ and for
by

Let us define the associated active sets 

\[ \mathcal{H}_i(\nu) = \gamma_c \int_{\Gamma_c} b_i p(\nu) \, ds - \sigma_i u_i - \max \left\{ 0, \gamma_c \int_{\Gamma_c} b_i p(\nu) \, ds - \sigma_i u_{bi} \right\} \]

\[ - \min \left\{ 0, \gamma_c \int_{\Gamma_c} b_i p(\nu) \, ds - \sigma_i u_{ai} \right\} \text{ a.e. in } [0, T], \]

\[ \mathcal{H}_{m+1}(\nu) = -\frac{\sigma_w}{\varepsilon} w - \frac{\vartheta}{\varepsilon} - \max \left\{ 0, \frac{\sigma_w}{\varepsilon^2} (y(\nu) - y_b) - \frac{\vartheta}{\varepsilon} \right\} \]

\[ - \min \left\{ 0, \frac{\sigma_w}{\varepsilon^2} (y(\nu) - y_a) - \frac{\vartheta}{\varepsilon} \right\} \text{ a.e. in } Q, \]

\[ \mathcal{H}_{m+2}(\nu) = \vartheta - \max \left\{ 0, \frac{\sigma_w}{\varepsilon} (w - w_b) \right\} \]

\[ - \min \left\{ 0, \frac{\sigma_w}{\varepsilon} (w - w_a) \right\} \text{ a.e. in } Q \]

for \( \nu = (u, w, \vartheta) \in \mathbb{Z} \times \mathbb{W} \) with \( y(\nu) = \bar{y} + Su \) and \( p(\nu) = \bar{p} + A_1 u - \frac{1}{\varepsilon} A_2 (\sigma w + \vartheta) \).

Then, we define

\[ \mathcal{G} = (\mathcal{H}_1, \ldots, \mathcal{H}_{m+2})^\top : \mathbb{Z} \times \mathbb{W} \to \mathbb{Z} \times \mathbb{W}. \]

Clearly, the nonsmooth operator equations (1.80), (1.82) and (1.84) become

\[ \mathcal{G}(\bar{\nu}) = 0 \text{ in } \mathbb{Z} \times \mathbb{W}, \]

where \( \bar{\nu} = (\bar{u}, \bar{w}, \bar{\vartheta}) \in \mathbb{Z} \times \mathbb{W}. \) Suppose that \( \nu = (u, w, \vartheta) \in \mathbb{Z} \times \mathbb{W} \) is an approximation for \( \bar{\nu}. \) According to (1.79) and (1.81), we set

\[ \alpha_i(\nu) = \gamma_c \int_{\Gamma_c} b_i p(\nu) \, ds - \sigma_i u_i \text{ for } i = 1, \ldots, m \text{ and } \beta(\nu) = -\frac{\sigma_w}{\varepsilon} w - \frac{\vartheta}{\varepsilon}. \]

Let us define the associated active sets

\[ A_{a_i}^U(\nu) = \left\{ t \in [0, T] \mid \alpha_i(\nu) + \sigma_i (u_i - u_{ai}) < 0 \text{ a.e.} \right\}, \quad i = 1, \ldots, m, \]

\[ A_{a_i}^U(\nu) = \left\{ t \in [0, T] \mid \alpha_i(\nu) + \sigma_i (u_i - u_{ai}) > 0 \text{ a.e.} \right\}, \quad i = 1, \ldots, m, \]

\[ A_a^W(\nu) = \left\{ (t, x) \in Q_T \mid \beta(\nu) + \frac{\sigma_w}{\varepsilon^2} (y(\nu) + \varepsilon w - y_a) < 0 \text{ a.e.} \right\}, \]

\[ A_b^W(\nu) = \left\{ (t, x) \in Q_T \mid \beta(\nu) + \frac{\sigma_w}{\varepsilon^2} (y(\nu) + \varepsilon w - y_b) > 0 \text{ a.e.} \right\}, \]

\[ A_{a_2}^W(\nu) = \left\{ (t, x) \in Q_T \mid w - w_a < 0 \text{ a.e.} \right\}, \]

\[ A_{b_2}^W(\nu) = \left\{ (t, x) \in Q_T \mid w - w_b > 0 \text{ a.e.} \right\}. \]
The associated inactive sets are defined as
\[ J_i^U(\nu) = [0,T] \setminus (A_{a_i}^U(\nu) \cup A_{b_i}^U(\nu)) \text{ for } i = 1, \ldots, m, \]
\[ J_i^W(\nu) = Q_T \setminus (A_{a_i}^W(\nu) \cup A_{b_i}^W(\nu)), \]
\[ J_2^W(\nu) = Q_T \setminus (A_{a_2}^W(\nu) \cup A_{b_2}^W(\nu)). \]

Throughout we denote by \( \chi \) the characteristic function of the set \( A \). Now, a particular Newton step is given by
\[ D\mathcal{G}(\nu)\nu^\delta = -\mathcal{G}(\nu) \text{ in } \mathbb{Z} \times \mathbb{W}, \]
where the Newton derivative \( D\mathcal{G} \) at \( \nu \) in direction \( \nu^\delta = (u^\delta, w^\delta, \vartheta^\delta) \in \mathbb{Z} \times \mathbb{W} \) is given as
\[ D\mathcal{G}(\nu)\nu^\delta = \left( \left( 1 - \chi_A^U(\nu) - \chi_A^W(\nu) \right) \gamma \int_{\Gamma_k} b_i p^\delta ds - \sigma_i u_i^\delta \right)_{1 \leq i \leq m} - \frac{\sigma_p}{2\varepsilon} \left( \varepsilon w^\delta + (\chi_{A_1}^W(\nu) + \chi_{A_2}^W(\nu)) y^\delta \right) - \frac{1}{\varepsilon} \left( \vartheta^\delta - (\chi_{A_1}^W(\nu) + \chi_{A_2}^W(\nu)) \vartheta^\delta \right), \]
where
\[ y^\delta = Su^\delta \text{ and } p^\delta = A_1 u^\delta - \frac{1}{\varepsilon} A_2 (\sigma_w w^\delta + \vartheta^\delta) \]
holds. The semismooth Newton method is summarized in Algorithm 1.

**Algorithm 1** (Semismooth Newton method with active sets)

1: Choose starting value \( \nu^0 = (u^0, w^0, \vartheta^0) \in \mathbb{Z} \times \mathbb{W} \) and set \( k = 0; \)
2: Determine \( y^0 = \tilde{y} + Su^0 \) and \( p^0 = \tilde{p} + A_1 u^0 - A_2 (\sigma_w w^0 + \vartheta^0)/\varepsilon. \)
3: Determine \( A_{a_i}^U(\nu^0), A_{b_i}^U(\nu^0), J_i^U(\nu^0) \) for \( i = 1, \ldots, m, A_{a_1}^W(\nu^0), A_{b_1}^W(\nu^0), J_1^W(\nu^0) \) and \( A_{a_2}^W(\nu^0), A_{b_2}^W(\nu^0) \) and \( J_2^W(\nu^0) \) from (1.85) and (1.86), respectively;
4: repeat
5: Compute the solution \( \nu^\delta = (u^\delta, w^\delta, \vartheta^\delta) \in \mathbb{Z} \times \mathbb{W} \) to
\[ D\mathcal{G}(\nu^k)\nu^\delta = -\mathcal{G}(\nu^k); \]
6: Set \( \nu^{k+1} = \nu^k + \nu^\delta \) and \( k = k + 1; \)
7: Determine \( y^k = \tilde{y} + Su^k \) and \( p^k = \tilde{p} + A_1 u^k - A_2 (\sigma_w w^k + \vartheta^k)/\varepsilon. \)
8: Determine \( A_{a_i}^U(\nu^k), A_{b_i}^U(\nu^k), J_i^U(\nu^k) \) for \( i = 1, \ldots, m, A_{a_1}^W(\nu^k), A_{b_1}^W(\nu^k), J_1^W(\nu^k) \) and \( A_{a_2}^W(\nu^k), A_{b_2}^W(\nu^k) \) and \( J_2^W(\nu^k) \) from (1.85) and (1.86), respectively;
9: if \( A_{a_i}^U(\nu^k) = A_{a_i}^U(\nu^{k-1}) \) and \( A_{b_i}^U(\nu^k) = A_{b_i}^U(\nu^{k-1}) \) for \( i = 1, \ldots, m \) then
1.3 The virtual control approach

10: if $A^W_a(\nu^k) = A^W_a(\nu^{k-1})$ and $A^W_b(\nu^k) = A^W_b(\nu^{k-1})$ then
11: if $A^W_{a,2}(\nu^k) = A^W_{a,2}(\nu^{k-1})$ and $A^W_{b,2}(\nu^k) = A^W_{b,2}(\nu^{k-1})$ then
12: flag = true;
13: end if
14: end if
15: end if
16: until flag = true;

Remark 1.47. Since the semi-smooth Newton method converges locally, it may happen that the initial guess $\nu^0$ is not close enough to the solution to guarantee convergence. Therefore, one should fix also a maximum number of iteration $K$ for Algorithm 1, checking if $G(\nu^K)$ is close to zero, i.e., if its norm is smaller than a given small tolerance $\tau$. If it is, then the solution $\nu^K$ is accepted, otherwise a different initial guess has to be chosen. Let us mention that this never happened in the numerical simulations shown in this thesis, due to the fact that the chosen initial control $u^0$ was enough to have convergence, but, as said, the possibility of convergence failure was taken in account in the numerical implementation. Moreover, another possible stopping criterion for Algorithm 1 is to check directly at each iteration if $\|G(\nu^k)\|_{Z \times W} \leq \tau$, but we do not include it in our algorithms.

Suppose that $\nu^k = (u^k, w^k, \vartheta^k) \in Z \times W, k \geq 0$, is a current iterate for Algorithm 1 and $(y^k, p^k)$ be given by step 3 of Algorithm 1. Moreover, $\nu^\delta = (u^\delta, w^\delta, \vartheta^\delta)$ denotes the solution to (1.88). Utilizing (1.87) and

\[ u^{k+1} = u^k + u^\delta, \quad w^{k+1} = w^k + w^\delta, \quad \vartheta^{k+1} = \vartheta^k + \vartheta^\delta, \quad y^{k+1} = y^k + y^\delta, \quad p^{k+1} = p^k + p^\delta, \]

we obtain that

\[ \gamma_c \int_{\Gamma_c} b_i^k p^{k+1} ds - \sigma_i u_i^{k+1} = 0 \quad \text{in} \ A^W_i(\nu^k), \quad i = 1, \ldots, m, \quad (1.89a) \]
\[ u_i^{k+1} = u_{ai} \quad \text{in} \ A^W_{ai}(\nu^k), \quad i = 1, \ldots, m, \quad (1.89b) \]
\[ u_i^{k+1} = u_{bi} \quad \text{in} \ A^W_{bi}(\nu^k), \quad i = 1, \ldots, m, \quad (1.89c) \]
\[ w^{k+1} = 0 \quad \text{in} \ A^W(\nu^k), \quad (1.89d) \]
\[ y^{k+1} + \varepsilon w^{k+1} = y_a \quad \text{in} \ A^W_a(\nu^k), \quad (1.89e) \]
\[ y^{k+1} + \varepsilon w^{k+1} = y_b \quad \text{in} \ A^W_b(\nu^k), \quad (1.89f) \]
\[ \vartheta^{k+1} = 0 \quad \text{in} \ A^W_2(\nu^k), \quad (1.89g) \]
\[ \sigma_w w^{k+1} - \varepsilon \vartheta^{k+1} = \sigma_w w_a \quad \text{in} \ A^W_{a,2}(\nu^k), \quad (1.89h) \]
\[ \sigma_w w^{k+1} - \varepsilon \vartheta^{k+1} = \sigma_w w_b \quad \text{in} \ A^W_{b,2}(\nu^k). \quad (1.89i) \]
Thus, $u_{i}^{k+1}$ is fixed on the active sets $\mathcal{A}_{ai}^{U}(\nu^{k})$ and $\mathcal{A}_{bi}^{U}(\nu^{k})$ for $i = 1, \ldots, m$. Analogously, $w^{k+1}$ is determined by $y^{k+1}$ on $\mathcal{A}_{a}^{W}$ and $\mathcal{A}_{b}^{W}$ and consequently $\dot{y}^{k+1}$ is determined by $y^{k+1}$ on the intersections of $\mathcal{A}_{a,2}^{W}$ and $\mathcal{A}_{b,2}^{W}$ with $\mathcal{A}_{a}^{W}$ and $\mathcal{A}_{b}^{W}$. These facts help to rewrite the state and adjoint equations as a unique coupled system. In fact, we infer from (1.81) and (1.89d)-(1.89i) that

$$
\beta^{k+1} = \begin{cases}
0 & \text{in } \mathcal{J}^{W}(\nu^{k}), \\
\frac{\sigma_{w}}{\varepsilon^{2}} (y^{k+1} - y_{a}) & \text{in } \mathcal{A}_{1}^{W}(\nu^{k}) := \mathcal{A}_{a}^{W}(\nu^{k}) \cap \mathcal{J}_{2}^{W}(\nu^{k}), \\
\frac{\sigma_{w}}{\varepsilon^{2}} (y^{k+1} - y_{b}) & \text{in } \mathcal{A}_{2}^{W}(\nu^{k}) := \mathcal{A}_{b}^{W}(\nu^{k}) \cap \mathcal{J}_{2}^{W}(\nu^{k}), \\
\frac{\sigma_{w}}{\varepsilon^{2}} \left(\frac{\varepsilon + 1}{\varepsilon} (y^{k+1} - y_{a}) + w_{a}\right) & \text{in } \mathcal{A}_{3}^{W}(\nu^{k}) := \mathcal{A}_{a}^{W}(\nu^{k}) \cap \mathcal{A}_{a,2}^{W}(\nu^{k}), \\
\frac{\sigma_{w}}{\varepsilon^{2}} \left(\frac{\varepsilon + 1}{\varepsilon} (y^{k+1} - y_{a}) + w_{b}\right) & \text{in } \mathcal{A}_{4}^{W}(\nu^{k}) := \mathcal{A}_{a}^{W}(\nu^{k}) \cap \mathcal{A}_{b,2}^{W}(\nu^{k}), \\
\frac{\sigma_{w}}{\varepsilon^{2}} \left(\frac{\varepsilon + 1}{\varepsilon} (y^{k+1} - y_{b}) + w_{a}\right) & \text{in } \mathcal{A}_{5}^{W}(\nu^{k}) := \mathcal{A}_{b}^{W}(\nu^{k}) \cap \mathcal{A}_{a,2}^{W}(\nu^{k}), \\
\frac{\sigma_{w}}{\varepsilon^{2}} \left(\frac{\varepsilon + 1}{\varepsilon} (y^{k+1} - y_{b}) + w_{b}\right) & \text{in } \mathcal{A}_{6}^{W}(\nu^{k}) := \mathcal{A}_{b}^{W}(\nu^{k}) \cap \mathcal{A}_{b,2}^{W}(\nu^{k}).
\end{cases}
$$

Inserting $\beta^{k+1}$ into the dual equation (c.f. (1.30)) we derive

$$
- \frac{d}{dt} \langle y^{k+1}(t), \varphi \rangle_{H} + a(t; \varphi, p^{k+1}(t)) + \sigma_{Q} \langle y^{k+1}(t), \varphi \rangle_{H}
+ \frac{\sigma_{w}}{\varepsilon^{2}} \left( y^{k+1}(t) \sum_{i=1}^{6} \chi_{A_{i}(\nu^{k})}(t) + \frac{1}{\varepsilon} y^{k+1}(t) \sum_{i=1}^{6} \chi_{A_{i}(\nu^{k})}(t), \varphi \right)_{H}
= \sigma_{Q} \langle y_{Q}(t), \varphi \rangle_{H} + \frac{\sigma_{w}}{\varepsilon^{2}} \left( r(\varepsilon, y_{a}, y_{b}, w_{a}, w_{b}, \chi_{A_{1}^{W}(\nu^{k})}, \ldots, \chi_{A_{6}^{W}(\nu^{k})})(t), \varphi \right)_{H}
$$

for all $\varphi \in V$ a.e. in $(0, T]$, where

$$
r(\varepsilon, y_{a}, y_{b}, w_{a}, w_{b}, \chi_{A_{1}^{W}(\nu^{k})}, \ldots, \chi_{A_{6}^{W}(\nu^{k})}) =
y_{a} \chi_{A_{1}^{W}(\nu^{k})} + y_{b} \chi_{A_{2}^{W}(\nu^{k})} - w_{a} \left( \chi_{A_{3}^{W}(\nu^{k})} + \chi_{A_{4}^{W}(\nu^{k})} \right) - w_{b} \left( \chi_{A_{5}^{W}(\nu^{k})} + \chi_{A_{6}^{W}(\nu^{k})} \right)
+ \frac{\varepsilon + 1}{\varepsilon} y_{a} \left( \chi_{A_{1}^{W}(\nu^{k})} + \chi_{A_{3}^{W}(\nu^{k})} \right) + \frac{\varepsilon + 1}{\varepsilon} y_{b} \left( \chi_{A_{2}^{W}(\nu^{k})} + \chi_{A_{4}^{W}(\nu^{k})} \right)
\quad \text{for all } \varphi \in V \text{ a.e. in } (0, T].
$$
Moreover, the terminal condition of the dual equation becomes

\[ p^{k+1}(T) = \sigma_T (y_T - y^{k+1}(T)) \]

\[ - \frac{\sigma_w}{\varepsilon^2} \left( y^{k+1}(T) \sum_{i=1}^{6} \chi_{A_i(\nu^k)}(T) + \frac{1}{\varepsilon} y^{k+1}(T) \sum_{i=3}^{6} \chi_{A_i(\nu^k)}(T) \right) \]

\[ + \frac{\sigma_w}{\varepsilon^2} r(\varepsilon, y_a, y_b, w_a, w_b, \chi_{A^w(\nu^k)}, \ldots, \chi_{A^w(\nu^k)})(T). \]

Furthermore, combining (1.79) and (1.89a)-(1.89c) we derive

\[ \alpha_i^{k+1} = \gamma_c \int_{\Gamma_c} b_i p^{k+1} \, ds - \sigma_i u_i^{k+1} = \begin{cases} 
0 & \text{in } \mathcal{J}^{il}_i(z^k), \\
\gamma_c \int_{\Gamma_c} b_i p^{k+1} \, ds - \sigma_i u_{ai} & \text{in } \mathcal{A}_a^{il}(z^k), \\
\gamma_c \int_{\Gamma_c} b_i p^{k+1} \, ds - \sigma_i u_{bi} & \text{in } \mathcal{A}_b^{il}(z^k) 
\end{cases} \]

and

\[ u_i^{k+1} = \begin{cases} 
\frac{\gamma_c}{\varepsilon} \int_{\Gamma_c} b_i p^{k+1} \, ds & \text{in } \mathcal{J}_i^{il}(z^k), \\
u_{ai} & \text{in } \mathcal{A}_a^{il}(z^k), \\
u_{bi} & \text{in } \mathcal{A}_b^{il}(z^k) 
\end{cases} \]

for \( i = 1, \ldots, m \). Inserting \( u^{k+1} \) into the state equation (1.3) we get

\[ \frac{d}{dt} \langle y^{k+1}(t), \varphi \rangle_H + a(y^{k+1}(t), \varphi) - \gamma_c \sum_{i=1}^{m} \chi_{A^{il}(z^k)}(t) \frac{\gamma_c}{\sigma_i} \int_{\Gamma_c} b_i p^{k+1}(t) d\tilde{s} \int_{\Gamma_c} b_i \varphi \, ds \]

\[ = \langle \mathcal{F}(t), \varphi \rangle_{V', V} + \gamma_c \sum_{i=1}^{m} \left( \chi_{A^{il}(z^k)}(t) u_{ai}(t) + \chi_{A^{il}(z^k)}(t) u_{bi}(t) \right) \int_{\Gamma_c} b_i \varphi \, ds, \]

\[ y^{k+1}(0) = y_0. \]

for all \( \varphi \in V \) a.e. in \( (0, T] \). Summarizing, the dual and primal equations can be formulated in the variables \( y^{k+1} \) and \( p^{k+1} \) only:

\[ \begin{pmatrix} A_1^k & A_2^k \\ A_2^k & A_2^k \end{pmatrix} \begin{pmatrix} y^{k+1} \\ p^{k+1} \end{pmatrix} = \begin{pmatrix} Q_1(\nu^k; u_a, u_b, b, \sigma, \gamma_c, \gamma_o, y_{out}) \\ Q_2(\nu^k; y_Q, y_T, y_a, y_b, w_a, w_b, \varepsilon, \sigma_w) \end{pmatrix}. \]  (1.95)

The operators \( A_1^k \) and \( A_2^k \) are of the form

\[ A_k^k = \mathcal{H} + \mathcal{A}_1^k, \quad A_{22}^k = \mathcal{H}^* + \mathcal{A}_{22}^k, \]
where \( \mathcal{H} \) stands for the heat-convection operator, which does not depend on \( k \). A discretization of (1.95) leads to the discrete system

\[
A = \begin{pmatrix}
A_{11}^k & A_{12}^k \\
A_{21}^k & A_{22}^k
\end{pmatrix}
\begin{pmatrix}
y^{k+1} \\
p^{k+1}
\end{pmatrix}
= \begin{pmatrix}
Q_1^k \\
Q_2^k
\end{pmatrix}.
\]

(1.96)

with

\[
A_{11}^k = H + \tilde{A}_{11}^k, \quad A_{22}^k = H^T + \tilde{A}_{22}^k,
\]

where \( H \) stands for the heat-convection operator, which is independent of \( k \). We resume the previous strategy in Algorithm 2.

**Algorithm 2** (Primal-dual active set strategy)

1. Choose starting value \( \nu^0 = (\nu^0, w^0, \theta^0) \in \mathcal{Z} \times \mathcal{W} \), set \( k = 0 \) and \( \text{flag} = \text{false} \);
2. Determine \( y^0 = \hat{y} + S \nu^0 \) and \( p^0 = \hat{p} + A \nu^0 - A_2(\sigma w^0 + \theta^0)/\varepsilon \);
3. Determine \( A_{ai}^u(\nu^0), A_{bi}^u(\nu^0), \tau_i^u(\nu^0) \) for \( i = 1, \ldots, m \), \( A_{ai}^w(\nu^0), A_{bi}^w(\nu^0), J^w(\nu^0) \) and \( A_{a2}^w(\nu^0), A_{b2}^w(\nu^0) \) from (1.85) and (1.86), respectively;
4: repeat
5: Compute the solution \( (y^{k+1}, p^{k+1}) \) by solving (1.96);
6: Compute \( \nu^{k+1} = (u^{k+1}, w^{k+1}, \theta^{k+1}) \in \mathcal{Z} \times \mathcal{W} \) from (1.89a)-(1.89i) and set \( k = k + 1 \);
7: Determine \( A_{a1}^u(\nu^k), A_{b1}^u(\nu^k), \tau_i^u(\nu^k) \) for \( i = 1, \ldots, m \), \( A_{a1}^w(\nu^k), A_{b1}^w(\nu^k), J^w(\nu^k) \) and \( A_{a2}^w(\nu^k), A_{b2}^w(\nu^k) \) from (1.85) and (1.86), respectively;
8: if \( A_{a1}^u(\nu^k) = A_{a1}^u(\nu^{k-1}) \) and \( A_{b1}^u(\nu^k) = A_{b1}^u(\nu^{k-1}) \) for \( i = 1, \ldots, m \) then
9: if \( A_{a1}^w(\nu^k) = A_{a1}^w(\nu^{k-1}) \) and \( A_{b1}^w(\nu^k) = A_{b1}^w(\nu^{k-1}) \) then
10: if \( A_{a2}^w(\nu^k) = A_{a2}^w(\nu^{k-1}) \) and \( A_{b2}^w(\nu^k) = A_{b2}^w(\nu^{k-1}) \) then
11: Set \( \text{flag} = \text{true} \);
12: end if
13: end if
14: end if
15: until \( \text{flag} = \text{true} \);

**Remark 1.48.** The discrete linear system (1.96) can be obtained discretizing the primal and dual equations (1.91)-(1.94), for example, with piecewise linear FE in space and implicit Euler scheme in time. Indicating the number of FE nodes with \( N_x \) and the number of time steps with \( N_t \), we have that the matrix \( A \in \mathbb{R}^{2N_xN_t \times 2N_xN_t} \) and
clearly it can be large enough to cause problems regarding the memory consumption and the computational time. Unfortunately, in our application, these problems cannot be avoided choosing a small $N_x$, since a coarse FE grid cannot capture the complexity of the dynamics involved. For example, as shown in Figure 1.1, with a small number of FE nodes we cannot capture all the vortices in the velocity field, missing important information. Our goal is then to apply POD to approximate the solution of (1.96) with a reduced-order system of dimension $2\ell N_t$ with $\ell \ll N_x$, gaining computational time but paying in approximation; cf. Chapter 2. On the other hand, also the number of time steps may not be small, since the time horizon $T$ could be extremely large, e.g., a month, and the heating system may have to interact frequently to control the temperature, e.g., every 10-15 minutes. It is also for this reason that we want to apply the MPC algorithm, where, solving many optimal control subproblems on a smaller time horizon, we can iteratively compute a feedback control that approximates the optimal control in $[0, T]$; cf. Chapter 3. Regarding the memory consumption, the matrix $A$ has a block structure that allows using a matrix-free GMRES algorithm to solve (1.96), although it is still necessary to store the finite element blocks which have dimension $N_x \times N_x$. Anyway, there is definitely an improvement in saving memory. In [64], the authors propose related preconditioners based on the block structure of matrix $A$. Since the blocks $A_{ii}^k$ for $i = 1, 2$ have themselves a sparse block structure, in Section 1.5.3 we apply a different preconditioner, i.e., we perform in advance the incomplete LU (ILU) factorization of each block in the principal diagonal of the matrix $H$, which contains the main information on the PDE dynamics. Then, we assemble a block diagonal preconditioner based on the previous ILU decomposition and we apply it in the
matrix-free GMRES iteration. Note that the expression matrix-free in our case means that we do not have to assemble the matrix $A$ and the preconditioner, but we use still the single FE blocks, e.g., the mass matrix. This approach let us to save some memory, which can be crucial if the number of FE degree of freedoms (DOFs) increases significantly; see Section 1.5.3.

1.4 The augmented Lagrangian method

Another way to avoid to deal directly with pointwise state constraints (and therefore measures) is to apply the augmented Lagrangian method, which is a penalty method based on an augmentation of the reduced cost functional $\tilde{J}(u)$ summing terms related to the pointwise state constraints. Following [77, 78], we define the slack variables $s_a := y - y_a$, $s_b := y_b - y$ a.e. in $Q_T$. Note that $s_a, s_b \in W$. Then, the inequality constraints (1.15c) can be equivalently rewritten as

$$y_a - y + s_a = 0 \quad \text{a.e. in } Q_T,$$

$$y - y_b + s_b = 0 \quad \text{a.e. in } Q_T,$$

(1.97)

(1.98)

together with the conditions $s_a \geq 0$ and $s_b \geq 0$ a.e. in $Q_T$. For $c > 0$ the augmented Lagrangian is defined as

$$L_c(y, u, s_a, s_b, \mu_a, \mu_b) = \tilde{J}(y, u) + \langle \mu_a, y_a - y + s_a \rangle_W + \langle \mu_b, y - y_b + s_b \rangle_W$$

$$+ \langle \mu_a(T), y_a(T) - y(T) + s_a(T) \rangle_H$$

$$+ \langle \mu_b(T), y(T) - y_b(T) + s_b(T) \rangle_H$$

$$+ \frac{c}{2} \|y_a - y + s_a\|_W^2 + \frac{c}{2} \|y - y_b + s_b\|_W^2$$

$$+ \frac{c}{2} \|y_a(T) - y(T) + s_a(T)\|_H^2$$

$$+ \frac{c}{2} \|y(T) - y_b(T) + s_b\|_H^2$$

with $\mu_a, \mu_b \in W$. Defining

$$X_{ad}^c = \{ x = (y, u, s_a, s_b) \in W(0, T) \times U \times W \times W \mid y = \hat{y} + S u, s_a \geq 0, s_b \geq 0 \text{ and } u \in U_{ad} \},$$

we can write the augmented Lagrangian problem as

$$\min L_c(y, u, s_a, s_b, \mu_a, \mu_b) \text{ s.t. } (y, u, s_a, s_b) \in X_{ad}^c$$

(1.99)
for \( c > 0 \) and \( \mu_a, \mu_b \in \mathcal{W} \). As in \([77, 78]\), to ensure the first-order optimality conditions

\[
\left\langle \frac{\partial L_c}{\partial s_a}(y, u, s_a, s_b, \mu_a, \mu_b), s - s_a \right\rangle_W \geq 0 \text{ for all } s \geq 0, s \in \mathcal{W},
\]

\[
\left\langle \frac{\partial L_c}{\partial s_b}(y, u, s_a, s_b, \mu_a, \mu_b), s - s_b \right\rangle_W \geq 0 \text{ for all } s \geq 0, s \in \mathcal{W},
\]

we impose

\[
s_a = \max \left\{ 0, y - y_a - \frac{\mu_a}{c} \right\} \geq 0, \quad (1.100)
\]

\[
s_b = \max \left\{ 0, y_b - y - \frac{\mu_b}{c} \right\} \geq 0. \quad (1.101)
\]

Hence, following \([77, 78]\) we can rewrite the augmented Lagrangian as

\[
L_c(y, u, \mu_a, \mu_b) = \mathcal{J}(y, u) - \frac{1}{2c} (\|\mu_a\|_W^2 + \|\mu_b\|_W^2 + \|\mu_a(T)\|_H^2 + \|\mu_b(T)\|_H^2) \\
+ \frac{c}{2} \max \left\{ 0, y_a - y_a + \frac{\mu_a}{c} \right\} \|_W^2 + \frac{c}{2} \max \left\{ 0, y - y_b + \frac{\mu_b}{c} \right\} \|_W^2 \\
+ \frac{c}{2} \max \left\{ 0, y_a(T) - y(T) + \frac{\mu_a(T)}{c} \right\} \|_H^2 \\
+ \frac{c}{2} \max \left\{ 0, y(T) - y_b(T) + \frac{\mu_b(T)}{c} \right\} \|_H^2.
\]

Thus, solving the optimal control problem (1.99) is equivalent to solve

\[
\min L_c(y, u, \mu_a, \mu_b) \text{ s.t. } y = \hat{y} + Su, u \in \mathcal{U}_{ad} \quad (L)
\]

for \( \mu_a, \mu_b \in \mathcal{W} \) and \( c > 0 \). Moreover, the corresponding reduced augmented Lagrangian is

\[
\hat{L}_c(u, \mu_a, \mu_b) := L_c(\hat{y} + Su, u, \mu_a, \mu_b) \\
= \mathcal{F}(u) - \frac{1}{2c} (\|\mu_a\|_W^2 + \|\mu_b\|_W^2 + \|\mu_a(T)\|_H^2 + \|\mu_b(T)\|_H^2) \\
+ \frac{c}{2} \max \left\{ 0, \hat{y}_a - Su + \frac{\mu_a}{c} \right\} \|_W^2 + \frac{c}{2} \max \left\{ 0, Su - \hat{y}_b + \frac{\mu_b}{c} \right\} \|_W^2 \\
+ \frac{c}{2} \max \left\{ 0, \hat{y}_a(T) - Su(T) + \frac{\mu_a(T)}{c} \right\} \|_H^2 \\
+ \frac{c}{2} \max \left\{ 0, Su(T) - \hat{y}_b(T) + \frac{\mu_b(T)}{c} \right\} \|_H^2.
\]
The optimal solution $\bar{u}$ of the reduced optimal control problem

$$\min \hat{L}_c(u, \mu_a, \mu_b) \text{ s.t. } u \in \mathcal{U}_{ad}$$

solves (L) with $\bar{y} := \hat{y} + \mathcal{S} \bar{u}$. Problem ($\hat{L}$) admits a unique solution for fixed $c > 0$ and $\mu_a, \mu_b \in \mathcal{W}$; cf. [92]. Hence, problem (L) admits a unique optimal solution $(\bar{y}, \bar{u})$. Clearly, the idea is to send progressively the penalty parameter $c$ to infinity and show convergence of the solution of (L) to the solution of (1.15). First questions that may arise are how large $c$ can be chosen at the begin and how much it has to increase at every iteration. For this reason, we follow the algorithm in [79, 80], where they propose an efficient way to deal with the initial choice of the penalty parameter $c$ and its increment. In [79, 80] the algorithm is applied to an elliptic state constrained optimal control problem requiring only an upper bound for the state. Our aim is to show that the same algorithm can be applied (and obviously adapted) to our parabolic bilateral state constrained optimal control problem. At first, we resume the proposed strategy in Algorithm 3, giving later all the necessary remarks and adapting to our setting the proof of convergence in [79, 80] in Section 1.4.2.

Algorithm 3 : First-order augmented Lagrangian method

1: **Data**: Initial pair $(\mu^0_a, \mu^0_b) \in \mathcal{W} \times \mathcal{W}$, initial weight $c_0 > 0$, increment $\beta > 0$ for $c_n$, tolerance $\epsilon > 0$, maximum number of iterations $n_{max}$, residuals $R^0_a, R^0_b \gg 1$, residual weight $\tau \in (0, 1)$;

2: set $n = 0$, $k = 0$ and $\text{flag} = \text{true}$;

3: while $\text{flag}$ and $n < n_{max}$ do

4: for fixed $(\mu^k_a, \mu^k_b)$ find $u^{k+1}$ solving the problem

$$\min \hat{L}_c(u, \mu^k_a, \mu^k_b) \text{ s.t. } u \in \mathcal{U}_{ad};$$

5: update the Lagrange multipliers

$$\mu^{k+1}_a = \max\{0, \mu^k_a + c_n(\hat{y}_a - \mathcal{S}u^{k+1})\};$$

$$\mu^{k+1}_b = \max\{0, \mu^k_b + c_n(\mathcal{S}u^{k+1} - \hat{y}_b)\};$$

6: compute the residuals

$$R^k_a = \|0, \hat{y}_a - S u^{k+1}\|_{C(\overline{\Omega})} + \max\{0, \int_0^T \langle \mu^{k+1}_a, Su^{k+1} - \hat{y}_a \rangle_H dt\}$$

$$+ \max\{0, \langle \mu^{k+1}_a(T), Su^{k+1}(T) - \hat{y}_a(T) \rangle_H\},$$

$$R^k_b = \|0, Su^{k+1} - \hat{y}_b\|_{C(\overline{\Omega})} + \max\{0, \int_0^T \langle \mu^{k+1}_b, \hat{y}_b - Su^{k+1} \rangle_H dt\}$$

$$+ \max\{0, \langle \mu^{k+1}_b(T), \hat{y}_b(T) - Su^{k+1}(T) \rangle_H\};$$
7: if \( R_a^k > \tau R_a^{n,+} \) \& \( R_b^k > \tau R_b^{n,+} \) then
8: reset \( \mu_{a,k+1} = \mu_{a,k} \) and \( \mu_{b,k+1} = \mu_{b,k} \);
9: reject \( u^{k+1} \);
10: increase penalty \( c_{k+1} = \beta c_k \);
11: else
12: accept \( u^{n+1,+} = u^{k+1} \) and the multipliers \((\mu_{a,n+1,+}, \mu_{b,n+1,+}) = (\mu_{a,c_{k+1}}, \mu_{b,c_{k+1}})\);
13: set \( c_{k+1} = c_k \);
14: set \( R_{a,n+1,+} = R_a^k \) and \( R_{b,n+1,+} = R_b^k \);
15: set \( n = n + 1 \);
16: end if
17: if \( R_{a,n,+} + R_{b,n,+} \leq \epsilon \) then
18: flag = false;
19: end if
20: set \( k = k + 1 \);
21: end while

**Definition 1.49.** If at iteration \( k \) we have that \( R_a^k \leq \tau R_a^{n,+} \) and \( R_b^k \leq \tau R_b^{n,+} \), we call the iteration successful. Otherwise, we call the iteration not successful.

To find the optimal control \( u \), we have replaced the problem \((\hat{P})\), which is a state and control constrained optimization problem, with the sequence of only control constrained problems \((\hat{L}_k)\) for \( k = 0, 1, \ldots \). Problem \((\hat{L}_k)\) can be solved utilizing a projected gradient algorithm or a BFGS method for fixed \( c_k > 0 \) and \( \mu_{a,c_k}, \mu_{b,c_k} \in W \); cf., e.g., [104]. Thus, in the next section, we compute the Riesz-representative of \( \hat{L}_c(u, \mu_a, \mu_b) \) and then we resume these optimization methods in Algorithms 4 and 5, respectively.

### 1.4.1 Gradient of the reduced augmented Lagrangian

In first place, one can note that

\[
\hat{L}_c(u, \mu_a, \mu_b)u^\delta = \hat{J}'(u)u^\delta + c \left\langle \max \left\{ 0, Su - \hat{y}_b + \frac{\mu_b}{c} \right\}, Su^\delta \right\rangle_W
- c \left\langle \max \left\{ 0, \hat{y}_a - Su + \frac{\mu_a}{c} \right\}, Su^\delta \right\rangle_W
+ c \left\langle \max \left\{ 0, Su(T) - \hat{y}_b(T) + \frac{\mu_b(T)}{c} \right\}, Su^\delta(T) \right\rangle_H
- c \left\langle \max \left\{ 0, \hat{y}_a(T) - Su(T) + \frac{\mu_a(T)}{c} \right\}, Su^\delta(T) \right\rangle_H.
\]

(1.102)
1. The linear-quadratic optimal control problem

Now, let $\mathcal{A}_4 : U \times W \times W \to W(0, T)$ be such that $p = \mathcal{A}_4(u, \mu_a, \mu_b)$ is the unique solution of

$$
-\frac{d}{dt} \langle p(t), \varphi \rangle_H + a(t; \varphi, p(t)) = c \left\langle \max \left\{ 0, \hat{y}_a(t) - Su(t) + \frac{\mu_a(t)}{c} \right\}, \varphi \right\rangle_H
$$

$$
- c \left\langle \max \left\{ 0, Su(t) - \hat{y}_b(t) + \frac{\mu_b(t)}{c} \right\}, \varphi \right\rangle_H,
$$

$$
p(T) = c \max \left\{ 0, \hat{y}_a(T) - Su(T) + \frac{\mu_a(T)}{c} \right\}
$$

$$
- c \max \left\{ 0, Su(T) - \hat{y}_b(T) + \frac{\mu_b(T)}{c} \right\},
$$

for all $\varphi \in V$ a.e. in $[0, T)$. To compute the gradient of the reduced augmented Lagrangian we can use the following lemma:

**Lemma 1.50.** Suppose that $u \in U_{ad}$, $y = \hat{y} + Su$ and $p = \hat{p} + \mathcal{A}_1 u + \mathcal{A}_4(u, \mu_a, \mu_b)$ for fixed $c > 0$ and $\mu_a, \mu_b \in W$. Moreover, $y^\delta = Su^\delta$ for $u^\delta \in U$. Then,

$$
\int_0^T \sigma_i \langle (y - y_Q)(t), y^\delta(t) \rangle_H \, dt
$$

$$
= \sigma_T \langle y_T - y(T), y^\delta(T) \rangle_H - \int_0^T \sum_{i=1}^m \gamma_c u_i^\delta(t) \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \, dt.
$$

**Proof.** This lemma can be proven similarly to Lemma 1.21. $\square$

Combining (1.18)-(1.20) with Lemma 1.50 and (1.102), we obtain

$$
\dot{\hat{L}}_c(u, \mu_a, \mu_b)u^\delta = \int_0^T \sum_{i=1}^m \left( \sigma_i u_i(t) - \gamma_c \langle b_i, p(t) \rangle_{L^2(\Gamma_c)} \right) u_i^\delta(t) \, dt
$$

(1.103)

for every $u^\delta \in U$ with $p = \hat{p} + \mathcal{A}_1 u + \mathcal{A}_4(u, \mu_a, \mu_b)$ and therefore the gradient $\nabla \dot{\hat{L}}_c(u, \mu_a, \mu_b) \in U$ – the Riesz representant of $\dot{\hat{L}}'_c(u, \mu_a, \mu_b)$ – reads

$$
\nabla \dot{\hat{L}}_c(u, \mu_a, \mu_b)(\cdot) = \left( \sigma_i u_i(\cdot) - \gamma_c \langle b_i, p(\cdot) \rangle_{L^2(\Gamma_c)} \right)_{1 \leq i \leq m}
$$

(1.104)

with $p = \hat{p} + \mathcal{A}_1 u + \mathcal{A}_4(u, \mu_a, \mu_b)$. Therefore, we can resume the first-order optimality conditions of problem $\dot{\hat{L}}_k$ in the following theorem:

**Theorem 1.51** (First-order optimality conditions for $\dot{\hat{L}}_k$). Let Assumptions 1.1 and 1.12 hold and suppose that the feasible set $U_{ad}$ is not empty. Let $u^k \in U_{ad}$ be the
optimal solution to \((\hat{L}_k)\) for a fixed \(c_k > 0\) and \(\mu_c^a, \mu_c^b \in \mathcal{W}\) with associated optimal state \(\bar{y}_k = \bar{y} + S\bar{u}^k\). Then, there exist unique Lagrange multipliers \(\bar{p}_k^a \in \mathcal{W}(0, T)\) and \((\bar{\alpha}_i^k)_{1 \leq i \leq m} \in \mathcal{U}\) satisfying the dual equations

\[
- \frac{d}{dt} \langle \hat{p}_k(t), \varphi \rangle_H + a(t; \varphi, \hat{p}_k(t)) = \langle \hat{\mu}_a^k - \bar{\mu}_a^k + \sigma_t(y_Q(t) - \bar{y}_k(t)), \varphi \rangle_H,
\]

\[
p_k(T) = \bar{\mu}_a^k(T) - \bar{\mu}_b^k(T) + \sigma_T(y_T - \bar{y}_k(T))
\]

(1.105a)

for all \(\varphi \in \mathcal{V}\) and a.e. in \([0, T)\) with

\[
\hat{\mu}_a^k := \max \{0, \mu_c^a + c_k(y_a - \bar{y}_k)\}, \quad \hat{\mu}_b^k := \max \{0, \mu_c^b + c_k(y_b - y_b)\}.
\]

(1.105b)

The Lagrange multipliers satisfy also the optimality system

\[
\sigma_i \bar{u}_i^k - \gamma_c \int_{\Gamma_i} b_i \hat{p}_k^k \, ds + \bar{\alpha}_i^k = 0 \text{ in } L^2(0, T) \text{ for } i = 1, \ldots, m
\]

(1.105c)

and

\[
\bar{\alpha}_i^k = \max \left\{0, \bar{\alpha}_i^k + \eta_i (\bar{u}_i^k - u_{bi})\right\} + \min \left\{0, \bar{\alpha}_i^k + \eta_i (\bar{u}_i^k - u_{ai})\right\}
\]

(1.105d)

for \(i = 1, \ldots, m\) and for arbitrarily chosen \(\eta_1, \ldots, \eta_m > 0\).

At last, for the sake of completeness, before proving that Algorithm 3 converges, we briefly resume the projected gradient method and the BFGS method in Algorithms 4 and 5, respectively.

**Algorithm 4** Projected Gradient Method for \((\hat{L}_k)\)

1: Choose an initial control \(u_0\), a maximum number of iterations \(k_{\text{max}}\) and a tolerance for the projected gradient \(\tau\);

2: Set \(k = 0\);

3: **repeat**

4: Compute the descend direction \(d^k = -\nabla L_c(u^k, \mu_a, \mu_b)\) from (1.104);

5: Compute the step size \(\tilde{s}_k\) through the Armijo Rule; cf. [104];

6: Compute the new control \(u^{k+1} = \text{Proj}_{\mathcal{U}_d} (u^k + \tilde{s}_k d^k)\);

7: Set \(k = k + 1\);

8: **until** \(\|u^{k+1} - u^k\|_u < \tau\) or \(k > k_{\text{max}}\)
Remark 1.52. Let us mention that with the operator $\text{Proj}_{U_{ad}}$ we indicate the projection operator onto the admissible set $U_{ad}$, i.e. given a control $u \in U$

$$\text{Proj}_{U_{ad}}(u_i)(t) = \begin{cases} u_{ai}(t) & \text{if } u_i(t) < u_{ai}(t), \\ u_{bi}(t) & \text{if } u_i(t) > u_{bi}(t), \\ u_i(t) & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, m$ and a.e. in $[0, T]$. 

\textbf{Algorithm 5} Projected BFGS Method for ($\hat{L}_k$)

1: Choose an initial control $u^0$, a maximum number of iterations $k_{\text{max}}$ and a tolerance for the projected BFGS $\tau$;
2: Set $k = 0$ and $H^0 = I_U$;
3: Compute the descend direction $d^0 = -\nabla \hat{L}_c(u^0, \mu_a, \mu_b)$ from (1.104)
4: repeat
5: Compute the step size $\tilde{s}_k$ through the Armijo Rule; cf. [104];
6: Set $p^k = \tilde{s}_k d^k$;
7: Compute the new control $u^{k+1} = \text{Proj}_{U_{ad}}(u^k + p^k)$;
8: Compute $\nabla \hat{L}_c(u^{k+1}, \mu_a, \mu_b)$ from (1.104);
9: Set $g^k = \nabla \hat{L}_c(u^{k+1}, \mu_a, \mu_b) - \nabla \hat{L}_c(u^k, \mu_a, \mu_b)$;
10: if $\langle p^k, g^k \rangle_U > 0$ then
11: Compute the new descend direction $d^{k+1} = -H^{k+1} \nabla \hat{L}_c(u^{k+1}, \mu_a, \mu_b)$, where the Hessian’s inverse approximation $H^{k+1}$ is obtained from
$$H^{k+1} = H^k + \frac{(p^k, g^k)_U + (g^k, H^k g^k)_U (p^k \otimes_U p^k) - (p^k, g^k)_U H^k g^k}{(p^k, g^k)_U} - \frac{H^k g^k \otimes_U p^k}{(p^k, g^k)_U}.$$ 

The product $\otimes_U$ is defined as
$$u \otimes_U v := \begin{pmatrix} \int_0^T u_1(t)v_1(t) \, dt & \cdots & \int_0^T u_1(t)v_m(t) \, dt \\ \vdots & \ddots & \vdots \\ \int_0^T u_m(t)v_1(t) \, dt & \cdots & \int_0^T u_m(t)v_m(t) \, dt \end{pmatrix},$$

for all $u, v \in U$; c.f., e.g., [104];
12: else
13: Set $d^{k+1} = -\nabla \hat{L}_c(u^{k+1}, \mu_a, \mu_b)$ and $H^{k+1} = I_U$;
14: end if
15: Set $k = k + 1$;
16: until $\|u^k - u^{k-1}\|_U < \tau$ or $k > k_{\text{max}}$
1.4.2 Proof of convergence

At first, we have to prove the following lemma, which justifies the choice for the residuals $R^k_a$ and $R^k_b$.

**Lemma 1.53.** Let $(\bar{y}, \bar{u})$ be the unique optimal solution to (1.15) and $\bar{\mu}_a, \bar{\mu}_a \in \mathcal{M}(\mathcal{Q})$ be the optimal Lagrange multiplier for problem (1.15), satisfying (1.17c)-(1.17d). Furthermore, let $(\hat{y}^k, \hat{u}^k)$ be the unique optimal solution to (L) for fixed $c_k > 0$ and $\mu_{a_k}, \mu_{b_k} \in W$. Then, it holds

$$\sigma_Q\|\bar{y} - \hat{y}^k\|^2_{L^2(0,T;H)} + \sigma_T\|\bar{y}(T) - \hat{y}^k(T)\|^2_H + \sum_{i=1}^m \sigma_i\|\bar{u}_i - \hat{u}_i^k\|^2_{L^2(0,T)} \leq$$

$$\int_0^T \langle \mu_a^k, \bar{y}^k - y_a \rangle_H \, dt + \int_\mathcal{Q} (y_a - \bar{y}^k) \, d\bar{\mu}_a$$

$$+ \langle \mu_b^k(T), \bar{y}^k(T) - y_a(T) \rangle_H + \langle \mu_b^k(T), y_b(T) - \bar{y}^k(T) \rangle_H$$

$$+ \int_0^T \langle \mu_b^k, y_b - \bar{y}^k \rangle_H \, dt + \int_\mathcal{Q} (\bar{y}^k - y_b) \, d\bar{\mu}_b$$

(1.106)

with $\bar{\mu}_a, \bar{\mu}_b^k \in W$ defined as in (1.105b).

**Proof.** At first, considering (1.3), we have that $\bar{y} - \hat{y}^k$ satisfies

$$\int_0^T \frac{d}{dt} \langle (\bar{y} - \hat{y}^k)(t), \varphi(t) \rangle_H + a(t; \bar{y} - \hat{y}^k, \varphi) \, dt = \int_0^T \langle B(\bar{u} - \hat{u}^k)(t), \varphi(t) \rangle_{V',V} \, dt$$

for all $\varphi \in W(0,T)$. Furthermore, let $\bar{p} = \hat{p} + \mathcal{A}_1 \bar{u} \in W(0,T)$ and $\hat{p}^k = \hat{p} + \mathcal{A}_1 \hat{u}^k + \mathcal{A}_4(\bar{u}^k, \mu_{a^k}, \mu_{b^k}) \in W(0,T)$. We recall that the operators $\mathcal{A}_1 : \mathcal{U} \to W(0,T)$ and $\mathcal{A}_4 : \mathcal{U} \times W \times W \to W(0,T)$ have been introduced in Remark 1.20 and in Section 1.4.1, respectively. From Remark 1.20 and (1.105a) we obtain

$$\int_0^T \frac{d}{dt} \langle (\bar{p} - \hat{p}^k)(t), \varphi(t) \rangle_H + a(t; \varphi, \bar{p} - \hat{p}^k) \, dt =$$

$$\int_0^T \langle \sigma_Q(\bar{y}^k - \bar{y})(t) + \bar{\mu}_b^k(t) - \bar{\mu}_b^k(t), \varphi(t) \rangle_H \, dt$$

for all $\varphi \in W(0,T)$. Choosing $\varphi = \bar{p} - \hat{p} \in W(0,T)$ in the first and $\varphi = \bar{y} - \hat{y}^k \in W(0,T)$ for the second and considering the respective values of $\bar{p}(T)$ and $\hat{p}^k(T)$, we
have

\[ \int_0^T \langle \mathcal{B}(\bar{u} - \bar{u}^k), \bar{p}^k - \bar{p} \rangle_{V', V} \, dt = \int_0^T \frac{d}{dt} \langle \bar{y} - \bar{y}^k, \bar{p}^k - \bar{p} \rangle_H + a(t; \bar{y} - \bar{y}^k, \bar{p}^k - \bar{p}) \, dt \]
\[ = \int_0^T \frac{d}{dt} \langle \bar{p}^k - \bar{p}, \bar{y} - \bar{y}^k \rangle_H + a(t; \bar{y} - \bar{y}^k, \bar{p}^k - \bar{p}) \, dt \]
\[ + \langle \bar{p}^k(T) - \bar{p}(T), \bar{y}(T) - \bar{y}^k(T) \rangle_H \]
\[ = \int_0^T \langle \sigma_Q(\bar{y} - \bar{y}^k) + \bar{\mu}_a^k - \bar{\mu}_b^k, \bar{y} - \bar{y}^k \rangle_H \, dt \]
\[ + \langle \sigma_T(\bar{y}(T) - \bar{y}^k(T) + \bar{\mu}_a^k(T) - \bar{\mu}_b^k(T), \bar{y}(T) - \bar{y}^k(T) \rangle_H \]

Applying a Lagrangian-based Adjoint approach [67], the variational inequality for problem (1.15) defined in (1.55) can be equivalently rewritten as

\[ \left\langle \begin{pmatrix} \sigma_1 \bar{u}_1 \\ \vdots \\ \sigma_m \bar{u}_m \end{pmatrix} - \mathcal{B}^* \bar{p}, u - \bar{u} \right\rangle_u - \int_Q S(u - \bar{u})d\bar{\mu}_a + \int_Q S(u - \bar{u})d\bar{\mu}_b \geq 0 \quad (1.107) \]

for all \( u \in \mathcal{U}_{ad} \). Similarly, the one for problem (L) is

\[ \left\langle \begin{pmatrix} \sigma_1 \bar{u}_1^k \\ \vdots \\ \sigma_m \bar{u}_m^k \end{pmatrix} - \mathcal{B}^* \bar{p}^k, u - \bar{u}^k \right\rangle_u \geq 0 \quad \text{for all } u \in \mathcal{U}_{ad}. \quad (1.108) \]

Then, testing (1.107) and (1.108) with \( u = \bar{u}^k \) and \( u = \bar{u} \) respectively, after some computations, one can obtain that

\[ \int_0^T \langle \mathcal{B}(\bar{u} - \bar{u}^k), \bar{p}^k - \bar{p} \rangle_{V', V} \, dt \leq -\sum_{i=1}^m \sigma_i \| \bar{u}_i - \bar{u}_i^k \|^2_{L^2(0, T)} + \int_Q (\bar{y} - \bar{y}^k)d\bar{\mu}_a 
\[ - \int_Q (\bar{y} - \bar{y}^k)d\bar{\mu}_b. \]
Hence, putting everything together, we have

\[ \sum_{i=1}^{m} \sigma_i \| \bar{u}_i - \bar{u}_i^k \|_{L^2(0,T)}^2 + \sigma_Q \| \bar{y} - \bar{y}^k \|_{L^2(0,T;H)}^2 + \sigma_T \| \bar{y}(T) - \bar{y}^k(T) \|_H^2 \leq \]

\[ \int_0^T \langle \bar{\mu}_a^k(t), \bar{y}^k(t) - \bar{y}(t) \rangle_H \, dt + \int_Q (\bar{y} - \bar{y}^k) \, d\bar{\mu}_a + \int_Q (\bar{y} - \bar{y}^k) \, d\bar{\mu}_b \]

Now, we show the boundedness of few terms on the right-hand side, the others can be estimated in the same way:

\[ \int_0^T \langle \bar{\mu}_a^k(t), \bar{y}^k(t) - \bar{y}(t) \rangle_H \, dt = \int_0^T \left[ \int_0^T \langle \bar{\mu}_a^k(t), \bar{y}^k(t) - y_a + y_a - \bar{y}(t) \rangle_H \, dt \right] \]

\[ \leq \int_0^T \langle \bar{\mu}_a^k(t), \bar{y}^k(t) - y_a \rangle_H \, dt \]

\[ \int_Q (\bar{y} - \bar{y}^k) \, d\bar{\mu}_a = \int_Q (\bar{y} - y_a) \, d\bar{\mu}_a + \int_Q (y_a - \bar{y}^k) \, d\bar{\mu}_a \]

These estimates come from the optimality conditions (1.17) and (1.105), in particular from (1.17c) and (1.105b). It is now clear, how one can prove the claim.

\[ \square \]

**Remark 1.54.** Following the idea of [79, 80], from Lemma 1.53 follows that

\[ \sum_{i=1}^{m} \sigma_i \| \bar{u}_i - \bar{u}_i^k \|_{L^2(0,T)}^2 + \sigma_Q \| \bar{y} - \bar{y}^k \|_{L^2(0,T;H)}^2 + \sigma_T \| \bar{y}(T) - \bar{y}^k(T) \|_H^2 \leq \]

\[ \max \left\{ 0, \int_0^T \langle \bar{\mu}_a^k(t), \bar{y}^k(t) - y_a \rangle_H \, dt \right\} + \| \bar{\mu}_a \|_{M(\overline{Q})} \| 0, y_a - \bar{y}^k \|_{C(\overline{Q})} \]

\[ + \max \left\{ 0, \langle \bar{\mu}_a^k(T), \bar{y}^k(T) - y_a(T) \rangle_H \right\} + \max \left\{ 0, \langle \bar{\mu}_b^k(T), y_b(T) - \bar{y}^k(T) \rangle_H \right\} \]

\[ + \max \left\{ 0, \int_0^T \langle \bar{\mu}_b^k(t), y_b - \bar{y}^k(t) \rangle_H \, dt \right\} + \| \bar{\mu}_b \|_{M(\overline{Q})} \| 0, \bar{y}^k - y_b \|_{C(\overline{Q})} . \]

Hence, when \( R_a^k \) and \( R_b^k \) become small, also the quantities on the right-hand side of the previous inequality get small as well. \[ \diamond \]
Now that is clear the reason to choose $R^k_a$ and $R^k_b$ as in Algorithm 3, we prove that $(\hat{y}^k, \hat{u}^k)$ converges to the solution $(\hat{y}, \hat{u})$ of (1.15) as the penalty parameter $c_k$ increases, where $\hat{y}^k = \hat{y} + Su^k$ and $\hat{u}^k$ is the solution to (\tilde{L}_k) for fixed $c_k$-independent multipliers $\mu_a, \mu_b \in \mathcal{W}$. In other words, we prove the following lemma:

Lemma 1.55. Let $\{c_k\}_{k \in \mathbb{N}}$ be a sequence of positive real numbers with $c_k \to +\infty$ as $k \to \infty$ and suppose that Assumptions 1.1, 1.12 and 1.14 hold true. Let $\mu_a, \mu_b \in \mathcal{W}$ with $\mu_a, \mu_b \geq 0$ be given and $(\hat{u}^k, \hat{p}^k)$ be respectively the solutions of (\tilde{L}_k) and (1.105a) for $\hat{p}^k_a \equiv \mu_a$ and $\hat{p}^k_b \equiv \mu_b$. Furthermore, $\hat{y}^k = \hat{y} + Su^k$. Then, it holds $(\hat{y}^k, \hat{u}^k) \to (\hat{y}, \hat{u})$ in $(W(0,T) \cap C(Q)) \times \mathcal{U}$ as $k \to \infty$, where $(\hat{y}, \hat{u})$ is the optimal solution to (1.15). Moreover, $r^k \to 0$ as $k \to \infty$ for $\hat{p}^k_a, \hat{p}^k_b$ defined as in (1.105b) and

$$r^k := \max \left\{ 0, \int_0^T \langle \hat{p}^k_a(t), \hat{y}^k(t) - y_s(t) \rangle_H dt \right\} + \max \left\{ 0, \langle \hat{p}^k_a(T), \hat{y}^k(T) - y_s(T) \rangle_H \right\} + \max \left\{ 0, \int_0^T \langle \hat{p}^k_b(t), y_s(t) - \hat{y}^k(t) \rangle_H dt \right\} + \max \left\{ 0, \langle \hat{p}^k_b(T), y_s(T) - \hat{y}^k(T) \rangle_H \right\}.$$

Proof. From the proof of Lemma 1.53, it holds

$$0 \leq \sum_{i=1}^m \sigma_i \|\hat{u}_i - \hat{u}^k_i\|_{L^2(0,T)}^2 + \sigma_Q \|\hat{y} - \hat{y}^k\|_{L^2(0,T;H)}^2 + \sigma_T \|\hat{y}(T) - \hat{y}^k(T)\|_H^2 \leq \int_0^T \langle \hat{p}^k_a(t) - \hat{p}^k_b(t), \hat{y}^k(t) - \hat{y}(t) \rangle_H dt + \langle \hat{p}^k_a(T) - \hat{p}^k_b(T), \hat{y}^k(T) - \hat{y}(T) \rangle_H$$

$$+ \int_Q (\hat{y} - \hat{y}^k) d\hat{p}_a - \int_Q (\hat{y} - \hat{y}^k) d\hat{p}_b.$$  \hfill (1.109)

Now, we are interested in estimating the terms on right-hand side. At first, from (1.105b) we have

$$\int_0^T \langle \hat{p}^k_a(t), \hat{y}^k(t) - \hat{y}(t) \rangle_H dt = \int_0^T \langle \hat{p}^k_a(t), \frac{-\mu_a(t)}{c_k} + \hat{y}^k(t) - y_s(t) \rangle_H dt$$

$$+ \int_0^T \left\langle \hat{p}^k_a(t), y_s(t) - \hat{y}(t) + \frac{\mu_a(t)}{c_k} \right\rangle_H dt \leq -\frac{1}{c_k} \|\hat{p}^k_a\|_W^2 + \frac{1}{c_k} \int_0^T \langle \hat{p}^k_a(t), \mu_a(t) \rangle_H dt$$

$$+ \int_0^T \langle \hat{p}^k_a(t), y_s(t) - \hat{y}(t) \rangle_H dt.$$  \hfill (1.110)
Applying Cauchy-Schwarz and Young’s inequalities we get
\[ \int_0^T \langle \tilde{\mu}_a^k(t), \tilde{y}^k(t) - \bar{y}(t) \rangle_H \, dt \leq -\frac{1}{c_k} \| \tilde{\mu}_a^k \|_W^2 + \frac{1}{2c_k} (\| \tilde{\mu}_a^k \|_W^2 + \| \mu_a \|_W^2) \]
\[ = -\frac{1}{2c_k} \| \tilde{\mu}_a^k \|_W^2 + \frac{1}{2c_k} \| \mu_a \|_W^2. \] \tag{1.111a}

Similarly, we obtain
\[ \int_0^T \langle \tilde{\mu}_b^k(t), \tilde{y}^k(t) - \bar{y}(t) \rangle_H \, dt \leq -\frac{1}{2c_k} \| \tilde{\mu}_b^k \|_W^2 + \frac{1}{2c_k} \| \mu_b \|_W^2, \] \tag{1.111b}
\[ \langle \tilde{\mu}_a^k(t), \tilde{y}^k(T) - \bar{y}(T) \rangle_H \leq -\frac{1}{2c_k} \| \tilde{\mu}_a^k(T) \|_H^2 + \frac{1}{2c_k} \| \mu_a(T) \|_H^2, \] \tag{1.111c}
\[ \langle \tilde{\mu}_b^k(T), \tilde{y}^k(T) - \bar{y}(T) \rangle_H \leq -\frac{1}{2c_k} \| \tilde{\mu}_b^k(T) \|_H^2 + \frac{1}{2c_k} \| \mu_b(T) \|_H^2. \] \tag{1.111d}

Consider \( \tilde{z}^k = \tilde{y} - \tilde{y}^k \) and \( \tilde{\nu}^k = \bar{u} - \bar{u}^k \), \( \tilde{z}^k \) is a weak solution of the following equation
\[ z_t(t, x) - \lambda \Delta z(t, x) + v(t, x) \cdot \nabla z(t, x) = 0, \quad \text{a.e. in } Q, \]
\[ \lambda \frac{\partial z}{\partial n}(t, s) + \gamma_e z(t, s) = \gamma_c \sum_{i=1}^m \tilde{\nu}^k(t) b_i(s), \quad \text{a.e. on } \Sigma_e, \]
\[ \lambda \frac{\partial z}{\partial n}(t, s) + \gamma_o z(t, s) = 0, \quad \text{a.e. on } \Sigma_o, \]
\[ z(0, x) = 0, \quad \text{a.e. in } \Omega. \] \tag{1.112}

Since \( \bar{u}, \bar{u}^k \in \mathcal{U}_{ad} \), it holds true that \( \bar{u}, \bar{u}^k \in \mathcal{U}^s \) for all \( s \in [1, +\infty] \) and therefore \( \tilde{\nu}^k \in \mathcal{U}^s \) as well. Let us mention that the space \( \mathcal{U}^s \) is defined in (1.76). Moreover, since \( b_1, \ldots, b_m \in L^\infty(\Gamma_e) \) we have that \( \gamma_c \sum_{i=1}^m \bar{u}^k b_i, \gamma_e \sum_{i=1}^m \bar{u}^k b_i, \gamma_o \sum_{i=1}^m \tilde{\nu}^k b_i \in L^s(\Sigma_e) \) for all \( s \in [1, +\infty] \). In particular, they are in \( L^s(\Sigma_e) \) for \( s > d + 1 \). Therefore, we can apply Lemma 1.8 having that \( \bar{y}, \tilde{y}^k \) and \( \tilde{z}^k \) are bounded in \( W(0, T) \cap C(\overline{Q}) \). Moreover, it holds
\[ \| \bar{y} - \tilde{y}^k \|_{W(0,T)} + \| \bar{y} - \tilde{y}^k \|_{C(\overline{Q})} \leq C(s) \| \bar{u} - \tilde{u}^k \|_{W}, \] \tag{1.113}
with \( C \) independent from \( \bar{u} \) and \( \tilde{u}^k \). In addition, since the sequence \( \{ \tilde{u}^k \} \) is bounded in \( \mathcal{U}^s \) for \( s \in [1, +\infty) \). Thus, we can extract weakly converging subsequences \( \tilde{u}^k' \rightharpoonup \tilde{u} \) in \( \mathcal{U}^s \), in particular for \( s > d + 1 \). Applying Lemma 1.8, \( \tilde{y}^k' \) converges strongly to \( \bar{y} = \tilde{y} + S \tilde{u} \) in \( L^2(0, T; H) \) and \( C(\overline{Q}) \). Then, using (1.113) we derive
\[ \int_{\overline{Q}} (\bar{y} - \tilde{y}^k) \, d\tilde{\mu}_a \leq C(s) \| \tilde{\mu}_a \|_{L^2(\overline{Q})} \| \bar{u} - \tilde{u}^k \|_{W} \leq C(s) \| \tilde{\mu}_a \|_{L^2(\overline{Q})} \| u_b - u_a \|_{W}, \] \tag{1.114a}
\[
\int_Q (\tilde{y} - \tilde{y}^k) \, d\tilde{\mu}_b \leq C(s) \|\tilde{\mu}_b\|_{M(Q)} \|\tilde{u} - \tilde{u}^k\|_{U^*} \leq C(s) \|\tilde{\mu}_a\|_{M(Q)} \|u_b - u_a\|_{U^*}. \tag{1.114b}
\]

Inserting (1.111) and (1.114) in (1.109) we obtain that

\[
\frac{1}{2c_k} \|\tilde{\mu}_a^k\|_W^2 + \frac{1}{2c_k} \|\tilde{\mu}_b^k\|_W^2 + \frac{1}{2c_k} \|\tilde{\mu}_a^k(T)\|_H^2 + \frac{1}{2c_k} \|\tilde{\mu}_b^k(T)\|_H^2 \leq C \tag{1.115}
\]

with \(C = C(s, c_k, \mu_a, \mu_b, \bar{\mu}_a, \bar{\mu}_b, u_a, u_b)\) a positive bounded constant depending on \(s, c_k, \mu_a, \mu_b, \bar{\mu}_a, \bar{\mu}_b, u_a\) and \(u_b\). The boundedness of the multipliers \(\tilde{\mu}_a^k\) and \(\tilde{\mu}_b^k\) shown in (1.115) can be used for proving that \(\tilde{y}\) satisfies the state constraints. In fact, passing to the limit for \(k' \to \infty\) in the identities

\[
\frac{1}{c_{k'}} \|\tilde{\mu}_a^{k'}\|_W^2 = c_{k'} \left\|\max \left\{0, \frac{\mu_a}{c_{k'}} + y_a - \tilde{y}^{k'}\right\}\right\|_W^2, \tag{1.116a}
\]

\[
\frac{1}{c_{k'}} \|\tilde{\mu}_b^{k'}\|_W^2 = c_{k'} \left\|\max \left\{0, \frac{\mu_b}{c_{k'}} + \tilde{y}^{k'} - y_b\right\}\right\|_W^2, \tag{1.116b}
\]

\[
\frac{1}{c_{k'}} \|\tilde{\mu}_a^{k'}(T)\|_H^2 = c_{k'} \left\|\max \left\{0, \frac{\mu_a(T)}{c_{k'}} + y_a(T) - \tilde{y}^{k'}(T)\right\}\right\|_H^2, \tag{1.116c}
\]

\[
\frac{1}{c_{k'}} \|\tilde{\mu}_b^{k'}(T)\|_H^2 = c_{k'} \left\|\max \left\{0, \frac{\mu_b(T)}{c_{k'}} + \tilde{y}^{k'}(T) - y_b(T)\right\}\right\|_H^2, \tag{1.116d}
\]

we have that \(y_a \leq \tilde{y} \leq y_b\) in \(Q_T\), because \(\frac{1}{c_{k'}} \|\tilde{\mu}_a^{k'}\|_W \to 0\), \(\max \left\{0, \frac{\mu_a}{c_{k'}} + y_a - \tilde{y}^{k'}\right\}\) converges to \(\max \{0, y_a - \tilde{y}\}\) in \(W\) and the other quantities in (1.116) behave similarly. What remains to prove is that \(\tilde{u} = \tilde{u}\) and \(\tilde{y} = \tilde{y}\). To do so, we combine again (1.109) and (1.111) together with (1.17c)-(1.17d) obtaining

\[
\sum_{i=1}^m \sigma_i \|\tilde{u}_i - \tilde{u}_i^k\|_{L^2(0,T)}^2 \leq \frac{1}{2c_k} \|\mu_a\|_W^2 + \frac{1}{2c_k} \|\mu_b\|_W^2 + \frac{1}{2c_k} \|\mu_a(T)\|_H^2 + \frac{1}{2c_k} \|\mu_b(T)\|_H^2,
\]

\[
+ \int_Q (y_a - \tilde{y}^k) \, d\tilde{\mu}_a + \int_Q (\tilde{y}^k - y_b) \, d\tilde{\mu}_b.
\tag{1.117}
\]

Sending \(k' \to \infty\) we have

\[
0 \leq \lim_{k' \to \infty} \sum_{i=1}^m \sigma_i \|\tilde{u}_i - \tilde{u}_i^k\|_{L^2(0,T)}^2 \leq \int_Q (y_a - \tilde{y}) \, d\tilde{\mu}_a + \int_Q (\tilde{y} - y_b) \, d\tilde{\mu}_b \leq 0.
\]

Hence, \(\tilde{u}^{k'} \to \tilde{u}\) in \(U\), which implies \(\tilde{y}^{k'} \to \tilde{y}\) in \(W(0,T)\) (and in particular in \(L^2(0,T,H)\)) for (1.5). Therefore, for the uniqueness of the limit \(\tilde{u} = \tilde{u}\) and \(\tilde{y} = \tilde{y}\),
meaning that \( \bar{y}^k \to \bar{y} \) also in \( C(Q) \). Due to the fact that the optimal solution of (1.15) is unique, the limit is independent from the chosen subsequence, hence we have convergence for the whole sequences to the limit points \( \bar{u} \) and \( \bar{y} \). At last, the estimates

\[
\begin{align*}
\int_0^T \langle \tilde{\mu}^k_a(t), \bar{y}^k(t) - y_a(t) \rangle_H \, dt &\leq \frac{1}{2c_k} \| \mu_a \|^2_W, \\
\int_0^T \langle \tilde{\mu}^k_b(t), y_b(t) - \bar{y}^k(t) \rangle_H \, dt &\leq \frac{1}{2c_k} \| \mu_b \|^2_W, \\
\langle \tilde{\mu}^k_a(T), \bar{y}^k(T) - y_a(T) \rangle_H &\leq \frac{1}{2c_k} \| \mu_a(T) \|^2_H, \\
\langle \tilde{\mu}^k_b(T), y_b(T) - \bar{y}^k(T) \rangle_H &\leq \frac{1}{2c_k} \| \mu_b(T) \|^2_H
\end{align*}
\]  

(1.118)
can be proved following the same techniques used to derive (1.111). Therefore

\[
0 \leq \liminf_{k \to \infty} r^k \leq \limsup_{k \to +\infty} r^k \leq 0
\]

which proves the claim. \( \square \)

As also mentioned in [79, 80], it is important to prove that Algorithm 3 makes infinitely many successful iterations. In fact, if the number of successful iterations is finite, there exists an index \( m > 0 \) such that for all iterations \( k > m \) the quantities \( R^k_a \) and \( R^k_b \) are not decreasing enough. This implies that the algorithm can not exit the while loop and consequently can not converge. Fortunately, the proof of the following lemma shows that this is impossible.

**Lemma 1.56.** Let Assumptions 1.1, 1.12 and 1.14 hold true. Then, Algorithm 3 makes infinitely many successful iterations.

**Proof.** The proof follows exactly the proof in [79, 80]. We report it in this thesis only for the sake of completeness. Assume that Algorithm 3 does only a finite number of successful iterations, then, as already mentioned, there exists an index \( m > 0 \) such that for all iterations \( k > m \) these iterations are not successful. Therefore, it holds \( \mu^k_a = \mu^m_a \) and \( \mu^k_b = \mu^m_b \) for all \( k > m \). Moreover, \( R^k_a > \tau R^m_a > 0 \) and \( R^k_b > \tau R^m_b > 0 \). From Lemma 1.55 and the fact that \( \bar{y} \) satisfies (1.15c), we have that

\[
\lim_{k \to \infty} R^k_a + R^k_b = \| \max \{0, y_a - \bar{y} \} \|_{C(Q)} + \| \max \{0, \bar{y} - y_b \} \|_{C(Q)} = 0,
\]

that leads to a contradiction. \( \square \)
Now that all the technical details are set up, convergence for the generated control sequence \( u^{n,+} \) and the associated state \( y^{n,+} := \hat{y} + Su^{n,+} \) is shown in the following theorem:

**Theorem 1.57.** Let Assumptions 1.1, 1.12 and 1.14 hold true. The sequence \( \{(u^{n,+}, y^{n,+})\}_{n \in \mathbb{N}} \) of successful iterations in Algorithm 3 converges to the optimal solution \((\bar{u}, \bar{y})\) of (1.15) in \( U \times W(0,T) \cap C(Q) \) as \( n \to \infty \).

**Proof.** For Lemma 1.56, the algorithm makes infinitely many successful iterations, therefore

\[
0 = \lim_{n \to \infty} R_a^{n,+} + R_b^{n,+}
\]

\[
= \lim_{n \to \infty} \left( \| \max \{0, y_a - y_a^{n+1,+}\} \|_{C(Q)} + \max \left\{ 0, \int_0^T \langle \mu_a^{n+1,+}, y_a^{n+1,+} - y_a \rangle_H \, dt \right\} \right.
+ \left( \| \max \{0, y_b^{n+1,+} - y_b \} \|_{C(Q)} + \max \left\{ 0, \int_0^T \langle \mu_b^{n+1,+}, y_b^{n+1,+} - y_b \rangle_H \, dt \right\} \right).
\]

Together with Lemma 1.53, it implies

\[
0 \leq \lim_{n \to \infty} \sum_{i=1}^m \sigma_i \| \bar{u}_i - u^{n,+} \|_{L^2(0,T)}^2 \leq 0.
\]

Therefore, \( u^{n,+} \) converges to \( \bar{u} \) in \( U \). Consequently, (1.5) implies that \( y^{n,+} \) converges strongly to \( \bar{y} \) in \( W(0,T) \). Now, since \( u^{n,+} \in U_{ad} \) holds, \( u^{n,+} \) is bounded in \( L^\infty(0,T; \mathbb{R}^m) \) for all \( n > 0 \). Hence, \( \{u^{n,+}\} \) is a bounded sequence in \( U^s \) for all \( s \in [1, +\infty) \), in particular for \( s > d + 1 \). This implies that there exist weakly converging subsequences \( \{u^{n,+}_s\} \) which weakly converge to \( \bar{u} \) in \( U^s \) for \( s > d + 1 \). Since \( u^{n,+} \) converges strongly in \( U \) to the unique limit point \( \bar{u} \), then, independently from the subsequence taken, the entire sequence \( \{u^{n,+}\} \) converges weakly to \( \bar{u} \) in \( U^s \). Finally, Lemma 1.8 implies that \( y^{n,+} \) converges strongly to \( \bar{y} \) in \( C(Q) \).

**1.5 Numerical tests**

All the tests in this thesis have been made on a Notebook Lenovo ThinkPad T450s with Intel Core i7-5600U CPU @ 2.60GHz and 12GB RAM. The codes are
written in Python language and we use the tools of Firedrake, cf. [112], PETSc, cf. [11, 12], and SLEPc, cf. [62, 119], for our numerical computations. In order to have a fair comparison of the methods previously presented, when it is not differently specified, we always make use of the data that will be presented in this section. At first, we choose as domain \( \Omega = [0, 5] \times [0, 5] \subset \mathbb{R}^2 \) and we introduce a triangulation with \( N_x = 2257 \) nodes, as shown in Figure 1.2(b). To compute the numerical solution of state and adjoint equations we use the Finite Element method in space and implicit Euler method in time. Moreover, as can be seen in Figure 1.2(a), we place \( m = 4 \) controls on the boundary \( \partial \Omega \) according to the following shape functions

\[
\begin{align*}
  b_1(x) &= \begin{cases} 1 & \text{if } x \in \{0\} \times [0, 1] \\ 0 & \text{otherwise}, \end{cases} & b_2(x) &= \begin{cases} 1 & \text{if } x \in [1, 2] \times \{5\} \\ 0 & \text{otherwise}, \end{cases} \\
  b_3(x) &= \begin{cases} 1 & \text{if } x \in \{5\} \times [3, 4] \\ 0 & \text{otherwise}, \end{cases} & b_4(x) &= \begin{cases} 1 & \text{if } x \in [2, 3] \times \{0\} \\ 0 & \text{otherwise}, \end{cases}
\end{align*}
\]

Hence,

\[
\Gamma_c = \{ x \in \partial \Omega \mid \exists \ i \in \mathbb{N} \text{ s.t. } b_i(x) = 1, 1 \leq i \leq m \}, \quad \Gamma_o = \partial \Omega \setminus \Gamma_c.
\]

For the state equation (1.1), we choose \( \gamma_c = 1, \gamma_o = 0.03, \) as diffusion coefficient \( \lambda = 1, \) as initial temperature profile \( y_\circ(x) = 15 + \sin(2\pi x_1) \cos(2\pi x_2) \) and as outside

\[
\begin{align*}
  \Gamma_c &= \{ x \in \partial \Omega \mid \exists \ i \in \mathbb{N} \text{ s.t. } b_i(x) = 1, 1 \leq i \leq m \}, \\
  \Gamma_o &= \partial \Omega \setminus \Gamma_c.
\end{align*}
\]

Figure 1.2: Domain \( \Omega \) with the location of the boundary controls, the inflow and the outflow (left), together with its triangulation (right).
temperature \( y_{\text{out}}(t) = 15 - t \) for all \( t \in [0, T] \). Furthermore, we generate the time-dependent velocity field \( \mathbf{v}(t, \mathbf{x}) \) solving the following Navier-Stokes equation\(^2\)

\[
\begin{align*}
\mathbf{v}_t + (\mathbf{v} \cdot \nabla) \mathbf{v} - \nu \Delta \mathbf{v} &= -\nabla P \quad \text{in } Q, \\
\nabla \cdot \mathbf{v} &= 0 \quad \text{in } Q, \\
P &= 0 \quad \text{in } [0, T] \times \Gamma_{\text{ol}} = \{ x_1 = 5.0, x_2 \in [4, 5] \}, \\
\mathbf{v} &= \mathbf{\bar{v}} \quad \text{in } [0, T] \times \Gamma_{\text{il}} = \{ x_1 = 0.0, x_2 \in [0, 1] \}, \\
\mathbf{v} &= 0 \quad \text{in } \Gamma \setminus \Gamma_{\text{ol}} \cup \Gamma_{\text{il}}, \\
\mathbf{v}(0) &= 0 \quad \text{in } \Omega,
\end{align*}
\]

where \( P \) is the pressure of the air in the room, \( \nu = 0.01 \) and

\[
\mathbf{\bar{v}}(t, \mathbf{x}) = (4.5(4.0x_2(1 - x_2)), 0, 0).
\]

In this scenario, we have a constant in time inflow \( \mathbf{\bar{v}}(t, \mathbf{x}) \) of maximum magnitude 4.5 on the bottom left side of \( \Omega \), i.e., \( \Gamma_{\text{il}} \), represented in green in Figure 1.2(a). Moreover, there is an outflow on the top right part of the domain, i.e., \( \Gamma_{\text{ol}} \), which is in blue in Figure 1.2(a). In Figure 1.3, we show the magnitude of \( \mathbf{v}(t, \mathbf{x}) \) for some time instances.

![Figure 1.3: Velocity field \( \mathbf{v}, t = 1.0, 2.0, 4.0 \)](image)

**Remark 1.58.** This choice of data describes a room where there are three radiators \( u_2, u_3 \) and \( u_4 \) and a warm air heater \( u_1 \), of which we can control only the temperature. Moreover, there is the possibility for the air to flow out from the room, e.g., in a presence of a door. For simplicity, we suppose that the wall permeability \( \gamma_o \) is constant and the room is subjected to the same external temperature on each side. ♦

\(^2\)The code is courtesy of Dr. J. Andrej, our DFG-project partner at Christian-Albrechts-Universität zu Kiel and a detailed explanation of the code can be found in his Ph.D. thesis [6].
1.5 Numerical tests

1.5.1 Tracking term

In this section, we prescribe a target that has to be reached inside the state constraints. This turns out to be helpful in the optimization problem to keep the solution inside the bounds. However, the corresponding optimal controls are not the smallest which guarantee that the temperature does not violate the constraints, as will be shown in Section 1.5.2. We choose as target $y_Q(t, x) = y_T(x) = 17$ for all $x \in \Omega$, $t \in [0, 4]$ and we fix $\sigma_T = \sigma_Q = 1000$. Furthermore, we choose as pointwise state constraints the following quantities

$$y_a(t, x) = \begin{cases} 
14 + t & t \in [0, 1.82683], \\
q(t) & t \in (1.82683, 2.17222), \\
16 & t > 2.17222, 
\end{cases}$$

where 

$$q(t) = -0.0202027t^6 + 0.242131t^5 - 1.0464t^4$$

$$+ 1.92074t^3 - 1.49028t^2 + 1.40626t + 14.$$  \hspace{1cm} (1.120)

The bottom constraint $y_a \in C(\overline{Q})$ is shown in Figure 1.4 and it is chosen in such a way to simulate an increasing heating process. In fact, we can not expect that a feasible solution exists setting $y_a(t, x) = 16$ from the beginning, because of the initial state $y_0$ and the position of the controls, which can not act immediately on every point of the domain $\Omega$. For example, in the bottom right corner the effect of the controls relies only on the velocity field $v(t, x)$ and on the diffusion coefficient $\lambda$. Furthermore, we set as control constraints $u_{a0}(t) = 0$ and $u_{b0}(t) = 45$ and as initial control $u^0_i(t) = 20$ for all $t \in [0, 4]$. For the time discretization we use a

![Figure 1.4: Lower state constraint $y_a$](image)
step $\Delta t = 0.05$ and we apply both methods presented in this chapter. For the virtual control approach we set and keep fixed $\varepsilon = 10^{-5}$ and $w_b = -w_a = 10^9$, pointing out the fact that the bounds on the virtual control $w$ can be chosen sufficiently large, since they have to be imposed only for convergence reason; see Section 1.3.3 and in particular Lemma 1.38. Furthermore, we choose the initial $w^0 = \varphi^0 = 0$ in $W$. For the augmented Lagrangian method, instead, we set as initial penalty parameter $c_0 = 0.1$, an increment $\beta = 4.0$, a residual weight $\tau = 0.95$, a residual tolerance of $10^{-5}$ and a maximum number of iterations $n_{\text{max}} = 1000$. In Table 1.1, we report a comparison between the two methodologies, where for the ALM we make use once of the projected gradient method (PG) and once of the projected BFGS. As it can be expected, the ALM-PG is the slowest among the methods tested. This is due to the nature of the projected gradient method, which is really robust, convergence speaking, but it takes many iterations before reaching the optimal solution. Therefore, combining the ALM with a quasi-Newton method like BFGS improves the computational time of the 97%, because the number of inner iterations needed to converge is significantly reduced. Surprisingly, the ALM algorithm is immediately successful, i.e., despite the fact it starts with a small penalty parameter $c_0$, Algorithm 3 takes only 1 outer iteration to drop the residuals under the tolerance chosen. This is justified by the presence of the tracking term: the optimal solution of the minimization problem ($\hat{L}_k$) for $k = 0$ is close enough to the target to be already inside the state constraints range. The PDASS converges in a really small number of steps and therefore this method is the quickest among the one tested. The reason is that the PDASS algorithm is equivalent to a semi-smooth Newton method, thus it is super-linearly convergent. However, we have not a significant speed-up compared to the BFGS time because each PDASS iteration costs significantly more. In fact, for each BFGS iteration and each line-search step we need to solve $2N_t = 160$ linear systems with $N_x = 2257$ unknowns, instead for the PDASS we have to find the solution of a linear system of dimension $2N_t N_x$, that is way more costly to solve. The dimension of the PDASS iteration matrix has

<table>
<thead>
<tr>
<th>Method</th>
<th>$|u|_u$</th>
<th>rel.err.Q</th>
<th>rel.err.T</th>
<th>Out. It.</th>
<th>In. It.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALM-PG</td>
<td>88.25</td>
<td>0.0459</td>
<td>0.0056</td>
<td>1</td>
<td>433</td>
<td>4625 s</td>
</tr>
<tr>
<td>ALM-BFGS</td>
<td>88.09</td>
<td>0.0460</td>
<td>0.0058</td>
<td>1</td>
<td>17</td>
<td>145 s</td>
</tr>
<tr>
<td>VC-PDASS</td>
<td>88.23</td>
<td>0.0459</td>
<td>0.0056</td>
<td>–</td>
<td>5</td>
<td>89 s</td>
</tr>
</tbody>
</table>

**Table 1.1:** VC and ALM results for the tracking term test
to be taken in account also for memory consumption, that can be avoided using a matrix-free implementation, as it is shown in Section 1.5.3. In Figure 1.5, we plot the maximum, minimum and average values of the optimal solution for each time instance, together with the target $y_Q$ and the state constraints $y_a$ and $y_b$. As one can notice, the optimal solution is always inside the bounds for all the three methods test, meaning that this approach is successful. The fact that the target is not reached exactly in all the points at each time step is due to the cost functional weights and due to the choice of the boundary conditions. In fact, the outside temperature is moving the optimal trajectory away from the target. Moreover, the way the controls are placed in the domain does not allow an instantaneous reaction.

**Figure 1.5:** Comparison between optimal solution maximum, minimum and average values, target $y_Q$ and constraints $y_a$ and $y_b$. 
to this effect. Anyway, as one can see in Table 1.1, the relative errors
\[\text{rel}\_\text{err}_Q := \frac{\|y - y_Q\|_{L^2(0,T;H)}}{\|y_Q\|_{L^2(0,T;H)}}, \quad \text{rel}\_\text{err}_T := \frac{\|y(T) - y_T\|_H}{\|y_T\|_H},\]
show that this deviation from the target is negligible, especially for the last time step \(T\) where the error is only of the 0.56\%. Let us mention that the peaks appearing in the maximum temperature reported in Figure 1.5 are due to the fact that the maximum is located in different grid points at each time step. Especially at the begin, the maximum temperature is connected to the control values, therefore, as soon as a control decreases, there may be a sudden switch of the position and of the value of the maximum, causing such peaks. Figure 1.5 shows also how the optimal trajectories computed by both ALMs and VC-PDASS are similar, especially the

\begin{figure}[h]
\centering
\begin{subfigure}{0.45\textwidth}
\includegraphics[width=\textwidth]{alm-pg}
\caption{ALM-PG}
\end{subfigure}\hfill
\begin{subfigure}{0.45\textwidth}
\includegraphics[width=\textwidth]{alm-bfgs}
\caption{ALM-BFGS}
\end{subfigure}
\begin{subfigure}{0.45\textwidth}
\includegraphics[width=\textwidth]{vc-pdass}
\caption{VC-PDASS}
\end{subfigure}
\caption{Optimal controls computed by the ALMs and the VC.}
\end{figure}
ALM-PG and VC-PDAISS ones. This is also confirmed by the control norm reported in Table 1.1 and by Figure 1.6, where the optimal controls are plotted.

### 1.5.2 Economic optimal control

In this section, we treat the case where the cost functional depends explicitly only on the control variables, therefore we set $\sigma_Q = \sigma_T = 0$, which is possible according to Assumption 1.12. Let us point out that the proofs (and thus the results of convergence) contained in Sections 1.3 and 1.4 do not change because of this choice. Anyway, it is clear that the presence of state constraints is crucial in this setting, otherwise the minimization algorithm will always select the smallest possible value for all the controls at each time step. This kind of optimal control problem is known as economic optimal control problem, because we do not focus on bringing the trajectory to a target, but only on optimizing the economic performance of the process, so that we can compute the smallest control that allows the state solution to satisfy the bounds. In this section, we keep the same control and state constraints of the previous test, i.e. $u_a(t) = 0.0$, $u_b(t) = 45.0$, $y_a(t, x) = \begin{cases} 14 + t & t \in [0, 1.82683], x \in \Omega, \\ q(t) & t \in (1.82683, 2.17222), x \in \Omega, \\ 16 & t > 2.17222, x \in \Omega, \end{cases}$ where $q(t)$ is defined in (1.120) and $y_b(t, x) = 29$. For the augmented Lagrangian method, we start with the initial penalty parameter $c_0 = 0.1$, with an increment $\beta = 4.0$ and a residual weight $\tau = 0.95$. The maximum number of iterations for the projected gradient (and projected BFGS) is 1000 and a tolerance of $10^{-5}$ for the residuals. From Table 1.2 we deduce that, also for this test, the projected BFGS method performs better than the projected gradient one in terms of computational time and iterations. For our application, the reduction of computational time is crucial, because the aim is to get as fast as possible the optimal control to apply it in real time. Since the convergence criterion on the residual $R_a^k$ and $R_b^k$ is satisfied,

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Method & $\|u\|_U$ & $k$ & $n$ & avg. it. & time \\
\hline
Proj. Gradient & 73.42 & 69 & 40 & 162.3 & $> 2$ days \\
Proj. BFGS & 72.10 & 63 & 40 & 30.6 & $\simeq 7$ hours \\
\hline
\end{tabular}
\caption{Economic optimal control problem: augmented Lagrangian method}
\end{table}
both methods are able to find the solution of the problem. Due to the quasi-Newton behaviour of the BFGS algorithm, the BFGS optimal controls are smaller than the projected gradient ones. This is also due to the choice for the maximum number of iterations, which is reached for the last six outer PG iterations, introducing a truncation error. It is also important to point out that the norm of the controls is smaller than the norms of the optimal controls in Section 1.5.1. This is due to the economic choice for the cost functional: now we are computing the smallest control such that the optimal state respects the state constraints, without focusing on a tracking term. Unfortunately, the computational time increases significantly, because the penalization parameter $c$ is not large enough to get the optimal solution in one successful outer iteration. However, starting with a large penalty parameter from the begin would not fix the issue, since we lose the warm start effect typical of the ALM and, therefore, we would need many costly inner iterations to converge. In Table 1.3, we report the numerical results for different choices of the parameter $\varepsilon$ in the VC. As one can see, the number of state active points, i.e., the number of points $(t, x)$ in the time-space grid whose solution $y(t, x)$ violate the constraints, gets smaller as $\varepsilon$ decreases, since for a decreasing $\varepsilon$ the solution of the VC is converging to the solution of (1.15), as also proved in Corollary 1.43. At the same time the number of iterations and the optimal controls increase to compensate the effort needed to keep the optimal trajectory inside the bounds, even if these controls remain smaller than the ones computed in Section 1.5.1, as expected. In Table 1.3, it is also reported the convergence order of the method, which is $O(\sqrt{\varepsilon})$ according to Remark 1.46. Since we do not know a priori the optimal solution and computing it

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$|u|_U$</th>
<th>State Act. Points</th>
<th>It.</th>
<th>Time</th>
<th>Conv. Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>42.81</td>
<td>146200 (81%)</td>
<td>9</td>
<td>140 s</td>
<td>–</td>
</tr>
<tr>
<td>0.281170</td>
<td>52.62</td>
<td>103749 (57.5%)</td>
<td>10</td>
<td>155 s</td>
<td>0.67</td>
</tr>
<tr>
<td>0.158113</td>
<td>59.37</td>
<td>73705 (41%)</td>
<td>10</td>
<td>170 s</td>
<td>0.78</td>
</tr>
<tr>
<td>0.088914</td>
<td>63.34</td>
<td>48392 (27%)</td>
<td>12</td>
<td>290 s</td>
<td>0.65</td>
</tr>
<tr>
<td>0.05</td>
<td>66.23</td>
<td>25924 (14.4%)</td>
<td>13</td>
<td>335 s</td>
<td>0.82</td>
</tr>
<tr>
<td>0.028117</td>
<td>67.90</td>
<td>15327 (8.5%)</td>
<td>18</td>
<td>482 s</td>
<td>0.53</td>
</tr>
<tr>
<td>0.015811</td>
<td>69.29</td>
<td>7542 (4.2%)</td>
<td>32</td>
<td>795 s</td>
<td>0.71</td>
</tr>
<tr>
<td>0.008891</td>
<td>70.09</td>
<td>4311 (2.4%)</td>
<td>40</td>
<td>923 s</td>
<td>0.59</td>
</tr>
<tr>
<td>0.005</td>
<td>70.91</td>
<td>2242 (1.2%)</td>
<td>49</td>
<td>1196 s</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 1.3: VC-PDASS convergence rate for decreasing $\varepsilon$. 

for a really small $\varepsilon$ can be costly and numerically challenging, we approximate the convergence order through the following formula

\[
\text{Conv. Order} \approx \log_\kappa \frac{\|u_{\kappa \varepsilon} - u_\varepsilon\|_U}{\|u_\varepsilon - u_{\varepsilon/\kappa}\|_U},
\]

where $\kappa = 4\sqrt{10}$. For decreasing $\varepsilon$ the numerical convergence rate is sometimes better than the expected 0.5. This may point out that the convergence estimate can be improved or that the chosen $\varepsilon$ are still too big to compute a good approximation of the convergence rate with the approximation formula (1.121). To improve the results and to compute the optimal solution for smaller $\varepsilon$, one has to refine the space-time grid, because the pointwise error between the optimal solution for $\varepsilon = 0.005$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{Comparison between optimal solution maximum, minimum and average values, target $y_Q$ and constraints $y_a$ and $y_b$.}
\end{figure}
and the state constraints is smaller than the discretization approximation error corresponding to the chosen $N_x$ and $N_t$. Similar discretization and convergence rate issues are also observed in [83]. In Figure 1.7, we compare the state constraints with the maximum, minimum and average values of the optimal solution in each time step for ALM-PG, ALM-BFGS and VC-PDASS. As one can see, for a decreasing $\varepsilon$ the PDASS solution is approaching the ALM-BFGS one, which is the best approximation of the optimal solution among the three computed. As already mentioned, the chosen maximum number of iterations influences the ALM-PG solution, making it worse than the ALM-BFGS one.

**Remark 1.59.** With our application in mind, for $\varepsilon = 0.005$ the magnitude of the violation of the state bounds is negligible, due to the fact that the human body can not perceive such small temperature differences. Therefore, there is no need in our case to send $\varepsilon$ to zero, but we have to choose it in such a way that these violations can be neglected. This last remark justifies the future tests in Chapter 2 and in Chapter 3.

### 1.5.3 PDASS matrix-free implementation

In this section we perform the same test of Section 1.5.2 fixing $\varepsilon = 0.005$, exploiting the block structure of the PDASS iteration matrix $A$ of (1.96) and applying our matrix-free preconditioner. From Table 1.4, one can see that the matrix-free GMRES can solve large system without running out of memory. Let us mention that the number of iteration before restarting the GMRES algorithm is not crucial in the failed tests, since the program runs out of memory while it is assembling the iteration matrix $A$. To point out that it is crucial to find a good preconditioner for the linear system matrix, we report in Table 1.5 the number of iterations needed by GMRES to converge to the solution with ten random initial guesses and ten random initial controls $u^0$ for the PDASS algorithm. As one can see, when the matrix $A$ is not preconditioned the number of iterations is of order $10^3-10^4$.

<table>
<thead>
<tr>
<th>$N_x$</th>
<th>585</th>
<th>2257</th>
<th>8865</th>
<th>35137</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Matrix-Free</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Table 1.4:** Results for different number of DOFs and $N_t = 3000$ time steps for PDASS. (X = Out of memory, ✓ = Convergence)
1.5 Numerical tests

When we apply an ILU preconditioner to the assembled matrix $A$, the convergence performance for the GMRES method improves, but, as already seen, such technique requires a lot of memory making it inapplicable for large systems. Instead, applying the matrix-free preconditioner proposed in Remark 1.48 is convenient in terms of memory, but the performance is way too dependent on the structure of the problem. Since we use only the blocks in the main diagonal of $A$, we are missing the coupling terms in state and adjoint equations, which depends on the active sets $A^W_a, A^W_b, A^W_{a,2}, A^W_{b,2}$ and on the inactive sets $I^W$ and $I^U_i$ for $i = 1, \ldots, m$. In fact, these blocks are in the off-diagonals of $A$. For this reason, the quality of the proposed preconditioner (MFB-PC) is affected by the number of active and inactive points and thus by the control variables. This explains why the MFB preconditioned GMRES performs worse than the not preconditioned one for some runs; cf. Table 1.5. Therefore, in future work, we will investigate matrix-free preconditioners based on different techniques, like domain decomposition methods; cf. [18, 60].

<table>
<thead>
<tr>
<th>Run Nr.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>No-PC</td>
<td>14313</td>
<td>2960</td>
<td>5675</td>
<td>2537</td>
<td>9132</td>
<td>7653</td>
<td>12984</td>
<td>1950</td>
<td>925</td>
<td>877</td>
</tr>
<tr>
<td>MFB-PC</td>
<td>9729</td>
<td>235</td>
<td>853</td>
<td>2811</td>
<td>1041</td>
<td>3426</td>
<td>17230</td>
<td>156</td>
<td>875</td>
<td>55</td>
</tr>
<tr>
<td>ILU PC</td>
<td>354</td>
<td>89</td>
<td>125</td>
<td>81</td>
<td>222</td>
<td>251</td>
<td>393</td>
<td>127</td>
<td>56</td>
<td>39</td>
</tr>
</tbody>
</table>

**Table 1.5:** Iterations for different random $u$ and random initial guess for GMRES (relative residual: $10^{-6}$, $N_x$: 2257, $N_t = 80$)
1. The linear-quadratic optimal control problem
Chapter 2

Proper Orthogonal Decomposition for linear-quadratic OCP

In this chapter we apply the POD method to our problem. We use this model order reduction technique to gain computational time speed-up and memory saving. At first, we briefly give the necessary basic knowledge about this method, fixing also the notation. Subsequently, we show how the regularization approaches described in Chapter 1 can be combined with POD. Further, we derive a-posteriori error estimators to certify the quality of the POD approximation and to develop strategies for updating the POD basis. At the end of the chapter, through numerical simulations, we study the effectiveness and the tightness of such estimates in relation to the parameters of the problem.

2.1 Basics

The results presented in this section marked an important milestone for model order reduction using POD, therefore we briefly mention them in order to fix the notation. For a detailed excursus on the topic we refer, e.g., to [57, 68, 85]. Let $S$ be either the space $H$ or the space $V$. In $S$ we denote by $\langle \cdot , \cdot \rangle_S$ and $\| \cdot \| = \langle \cdot , \cdot \rangle_S^{1/2}$ the inner product and the associated norm, respectively. For fixed $K \in \mathbb{N}$ let the so-called snapshots $s^k \in L^2(0, T; S)$ be given for $1 \leq k \leq K$. We introduce the linear subspace $S^K \subset S$, where

$$s \in S^K \Leftrightarrow \exists \omega^1, \ldots, \omega^K \in L^2(0, T) : s = \sum_{k=1}^{K} \int_{0}^{T} \omega^k(t)s^k(t) \, dt. \quad (2.1)$$
We call the set $S^K$ the **snapshots subspace** and we indicate its dimension with $D \geq 1$. Let $\{\psi_i\}_{i=1}^D$ denote an orthonormal basis for $S^K$, then each snapshot can be expressed as

$$s^k(t) = \sum_{i=1}^D \langle s^k(t), \psi_i \rangle_S \psi_i, \quad \text{a.e. in } [0,T], \text{ for } k = 1, \ldots, K.$$  \hfill (2.2)

The POD method consists in choosing an orthonormal basis $\{\psi_i\}_{i=1}^D$ for $S^K$ such that for every $\ell \in \{1, \ldots, D\}$ the mean square error between the snapshots $s^k$ and their corresponding $\ell$-th partial sum of (2.2) is minimized:

$$\min_{\{\psi_i\}_{i=1}^\ell} \sum_{k=1}^K \int_0^T \| s^k(t) - \sum_{i=1}^\ell \langle s^k(t), \psi_i \rangle_S \psi_i \|^2_S \, dt$$  \hfill (2.3)

subject to $\{\psi_i\}_{i=1}^\ell \subset S$ and $\langle \psi_i, \psi_j \rangle_S = \delta_{ij}, 1 \leq i, j \leq \ell$,

where $\delta_{ij}$ is the Kronecker delta.

**Definition 2.1.** A solution $\{\psi_i\}_{i=1}^\ell$ to (2.3) is called a POD basis of rank $\ell$. We define the subspace spanned by the first $\ell$ POD basis functions as $S^\ell = \text{span}\{\psi_1, \ldots, \psi_\ell\}$.

Using a Lagrangian framework, the solution to (2.3) is characterized by the following optimality conditions:

$$\mathcal{R}\psi = \lambda\psi,$$  \hfill (2.4)

see [68, 85], where $\mathcal{R} : S \to S$ is given by

$$\mathcal{R}\psi = \sum_{k=1}^K \int_0^T \langle s^k(t), \psi \rangle_S s^k(t) \, dt \quad \text{for } \psi \in S.$$ 

It can be proven that $\mathcal{R}$ is a compact, nonnegative and self-adjoint operator. By Hilbert-Schmidt theory, see [118], there exists an orthonormal basis $\{\psi_i\} \in S$ and a sequence $\{\lambda_i\} \in \mathbb{R}$ of non-negative real numbers so that

$$\mathcal{R}\psi_i = \lambda_i\psi_i, \quad \lambda_1 \geq \cdots \geq \lambda_D > 0 \quad \text{and} \quad \lambda_i = 0, \quad \text{for } i > D.$$  \hfill (2.5)

Moreover $S^K = \text{span}\{\psi_i\}_{i=1}^D$. Notice that $\mathcal{R}$, $\{\lambda_i\} \in \mathbb{R}$ and $\{\psi_i\} \in S$ depend on $K$. Throughout the thesis we omit to indicate this dependency. Furthermore, the error formula

$$\sum_{k=1}^K \int_0^T \| s^k(t) - \sum_{i=1}^\ell \langle s^k(t), \psi_i \rangle_S \psi_i \|^2_S \, dt = \sum_{i=\ell+1}^D \lambda_i,$$  \hfill (2.6)

holds true for the POD basis $\{\psi_i\}_{i=1}^\ell$ of rank $\ell$; cf. [68].
Remark 2.2. a) In the context of optimal control problems a reasonable choice for the snapshots is, e.g., the solution of state and adjoint equations for an arbitrary admissible control.

b) For the numerical realization, the space \( S \) has to be discretized by, e.g., a finite element discretization in space and a finite difference one in time. In this case the Hilbert space \( S \) has to be replaced by a Euclidean space \( \mathbb{R}^{N_x} \) endowed with a weighted inner product. \( N_x \) corresponds to the number of FE nodes and we choose \( \ell \ll N_x \). Furthermore, we have to perform a trapezoidal approximation for the integral in time in (2.3); see [57, Chapter 1].

If a POD Basis \( \{ \psi_i \}_{i=1}^\ell \) of rank \( \ell \) is computed, we can derive a reduced-order model for (1.3): for any \( g \in L^2(0,T;\mathbb{R}^m) \) the function \( q^\ell = S^\ell g \) is given by

\[
\frac{d}{dt} \langle q^\ell(t), \psi \rangle_H + a(t; q^\ell(t), \psi) = \langle (B g)(t), \psi \rangle_{V',V} \quad \text{for all } \psi \in S^\ell \text{ a.e. in } (0,T],
\]

\[
q^\ell(0) = 0 \quad \text{in } H.
\]

(2.7)

For any \( u \in U_{ad} \) the POD approximation \( y^\ell \) for the state solution is \( y^\ell = \hat{y}^\ell + S^\ell u \), where \( \hat{y}^\ell \) solves

\[
\frac{d}{dt} \langle \hat{y}^\ell(t), \psi \rangle_H + a(t; \hat{y}^\ell(t), \psi) = \langle F(t), \psi \rangle_{V',V} \quad \text{for all } \psi \in S^\ell \text{ a.e. in } (0,T],
\]

\[
\hat{y}^\ell(0) = y_0 \quad \text{in } H.
\]

(2.8)

Analogously a reduced-order model can be derived for the adjoint equation; see, e.g., [57, Chapter 1]. The POD Galerkin approximation of (\( \hat{P} \)) is given by

\[
\min J^\ell(z) = \min J(\hat{y}^\ell + S^\ell u, z) \quad \text{s.t. } z \in Z_{ad}^\ell,
\]

(\( \hat{P}^\ell \))

where the set of admissible controls is

\[
Z_{ad}^\ell = \{ z = (u,w) \in Z \mid u \in U_{ad}, y_a \leq \hat{y}^\ell + (S^\ell u)(t,x) + \varepsilon w(t,x) \leq y_b, w_a \leq w \leq w_b \}.
\]

Similarly, the POD Galerkin approximation of (\( \hat{L} \)) is given by

\[
\min \hat{L}^\ell_{ca}(u,\mu_a^c,\mu_b^c) \text{ s.t. } u \in U_{ad},
\]

(\( \hat{L}^\ell \))
where the POD-based reduced augmented Lagrangian is defined as
\[
\hat{L}^\ell_c(u, \mu^c_a, \mu^c_b) := \hat{J}^\ell(u) - \frac{1}{2c} \left( \|\mu^c_a\|_{L^2(Q)}^2 + \|\mu^c_b\|_{L^2(Q)}^2 + \|\mu^c_a(T)\|_{L^2(\Omega)}^2 + \|\mu^c_b(T)\|_{L^2(\Omega)}^2 \right)
+ \frac{c}{2} \left\| \max \left\{ 0, \hat{y}_a - \mathcal{S}^\ell u + \frac{\mu^c_a}{c} \right\} \right\|_{L^2(Q)}^2
+ \frac{c}{2} \left\| \max \left\{ 0, \mathcal{S}^\ell u - \hat{y}_b + \frac{\mu^c_b}{c} \right\} \right\|_{L^2(Q)}^2
+ \frac{c}{2} \left\| \max \left\{ 0, \hat{y}_a(T) - (\mathcal{S}^\ell u)(T) + \frac{\mu^c_a(T)}{c} \right\} \right\|_{L^2(\Omega)}^2
+ \frac{c}{2} \left\| \max \left\{ 0, (\mathcal{S}^\ell u)(T) - \hat{y}_b(T) + \frac{\mu^c_b(T)}{c} \right\} \right\|_{L^2(\Omega)}^2,
\]
where
\[
\hat{J}^\ell(u) := J(\hat{y} + \mathcal{S}^\ell u, u).
\]

**Remark 2.3.** Since the PDASS is an all at once approach, in the POD formulation we have to reduce the state and the adjoint variables including the parts which do not depend on the controls, like \(\hat{y}\) and \(\hat{p}\), respectively. The reason relies on the fact that we can not consider the splitting of the variables \(y\) and \(p\) solving system (1.96). In the ALM method, instead, we can compute \(\hat{y}\) and \(\hat{p}\) in advance and build a reduced-order model only for the parts depending on the control variables.  

## 2.2 POD-based primal-dual active set strategy

As already mentioned in Remark 1.48, the PDASS algorithm can be really costly in terms of computational time, especially when the number of DOFs is large. Therefore, it is natural to apply a model order reduction approach to the problem. Once a POD basis \(\{\psi_i\}_{i=1}^\ell\) of rank \(\ell\) is computed, one can derive the following POD Galerkin formulation of the PDASS coupled equations (1.91)-(1.94).

The reduced-order state equation is
\[
\frac{d}{dt} \langle y^{k+1,\ell}(t), \psi \rangle_H + a(t; y^{k+1,\ell}(t), \psi) - \gamma_c \sum_{i=1}^m \chi_{\mathcal{A}^u_i}(z^k(t)) \gamma_c \int_{\Gamma_c} b_i \psi^{k+1,\ell}(t) d\tilde{s} \int_{\Gamma_c} b_i \psi \, ds
= \mathcal{F}(t, \psi)_{V^*,V} + \gamma_c \sum_{i=1}^m \left( \chi_{\mathcal{A}^u_i(z^k)}(t) u_{ai}(t) + \chi_{\mathcal{A}^u_i(z^k)}(t) u_{bi}(t) \right) \int_{\Gamma_c} b_i \psi \, ds
\]
\[(2.9)\]
2.2 POD-based primal-dual active set strategy

for all $\psi \in S^\ell$ a.e. in $[0, T]$. Similarly, the reduced-order dual equation reads

$$
\frac{d}{dt} \langle y^{k+1,\ell}(t), \psi \rangle_H + a(t; \psi, p^{k+1,\ell}(t)) + \sigma_Q \langle y^{k+1,\ell}(t), \psi \rangle_H
$$

$$
+ \frac{\sigma_w}{\varepsilon^2} \left( y^{k+1,\ell}(t) \sum_{i=1}^6 \chi_{A_i(\nu^k)}(t) + \frac{1}{\varepsilon} y^{k+1,\ell}(t) \sum_{i=3}^6 \chi_{A_i(\nu^k)}(t), \psi \right)_H
$$

$$
= \sigma_Q \langle y_Q(t), \psi \rangle_H + \frac{\sigma_w}{\varepsilon^2} \left( r(\varepsilon, y_a, y_b, w_a, w_b, \chi_{A_W^6(\nu^k)}(t), \dots, \chi_{A_W^6(\nu^k)}(t)), \psi \right)_H
$$

(2.10)

for all $\psi \in S^\ell$ a.e. in $[0, T]$ where $r(\cdot, \ldots, \cdot)$ is defined in (1.92). Likewise, $y^{k+1,\ell}(0)$ and $p^{k+1,\ell}(T)$ can be defined. When equation (1.91)-(1.94) are discretized with piecewise linear FE, the discrete FE matrices have dimension $N_x \times N_x$, where $N_x$ is the number of FE nodes. The respective reduced order matrices describing (2.9) and (2.10) belong to $\mathbb{R}^{\ell \times \ell}$ with $\ell \ll N_x$. Therefore, in the POD-based PDASS algorithm one has to solve only a linear system of dimension $2\ell N_t$ at each Newton iteration, where $N_t$ is the number of time steps. Hence, we gain a speed-up of the computational time. In particular, this happens in real application, where $\ell$ has an order of magnitude of $10^2 \sim 10^3$, while $N_x$ has one of $10^6 \sim 10^9$.

Remark 2.4. So far, some issues are still not clear: how to generate good snapshots (and therefore a good POD-basis), how to choose the rank $\ell$ and how to ensure the quality of the approximation. The first issue is strictly related to the problem structure itself. Therefore, there is no good recipe that works for every situation. In the context of optimal control problems, one can start from the idea that the best snapshots are the one related to the optimal solution. Clearly, if we are able to have an a-priori knowledge of the optimal solution, we do not need to solve the optimization problem. For the POD-PDASS algorithm, we suggest to compute some optimization steps with a cheap first-order method, e.g., the augmented Lagrangian method and then generate the POD-basis with the state and adjoint snapshots computed with the obtained sub-optimal control. Another option to get a sub-optimal control is to compute one or two steps directly with the full-order semi-smooth Newton method, if the computational time related to the application allows it. What we want to point out is that these procedures are all heuristic and related to the problem. For the second issue, one can choose the rank $\ell$ looking at the a-priori error estimate (2.6) and selecting the smallest $\ell$ such that

$$
\sum_{i=\ell+1}^D \lambda_i \leq \tau
$$

(2.11)
where $0 < \tau \ll 1$ is a predefined tolerance. Of course, this procedure guarantees that the snapshots $s^k$ are well reconstructed by the solution of the POD-Galerkin approximation, but, if the snapshots are still not good for the problem, the suboptimal POD solution $(u^\ell, w^\ell)$ may be far from the optimal solution $(\bar{u}, \bar{w})$. This leads to the third issue regarding the quality of the approximation, which can be measured with an a-posteriori error estimator, cf. Section 2.4. If the a-posteriori estimate is larger than a predefined tolerance, we could increase $\ell$ or generate a new basis from new snapshots. Also in this situation, the procedures to follow are heuristic, problem related and dependent on the tightness of the a-posteriori estimate, meaning that there is no unique recipe that efficiently solves each mentioned issue.

Based on the considerations done in Remark 2.4, we present the POD-PDASS method in Algorithm 6. To keep the algorithm simple, we suppose that few iterations $K$ of the Newton algorithm are sufficient to generate good snapshots. Moreover, we suppose that the tolerance $\tau_1$ is sufficiently small. These previous assumptions are made to guarantee that we do not update the POD basis after each step of the semi-smooth Newton method. If such assumptions do not hold, according to Remark 2.4, one can think to not only update the basis in Algorithm 6, but also to increase $\ell$. It may be significant also to choose a greater $K$ at the begin. Let us mention that, for example, in [14] and [45] a detailed overview on possible basis update strategies is presented, respectively, for a POD-based multi-objective and non-linear optimal control problems.

**Algorithm 6** (POD-based primal-dual active set strategy)

1: Choose starting value $\nu^0 = (u^0, w^0, \vartheta^0) \in Z \times W$, number of Newton steps to compute the snapshots $K$, tolerance $\tau_1$ for the a-priori estimate (2.11) and tolerance $\tau_2$ for the a-posteriori error estimate (2.14)
2: Set $k = 0$, flag = false and snapflag = true;
3: Determine $y^0 = \hat{y} + Su^0$ and $p^0 = \hat{p} + Au^0 - A_2(\sigma w^0 + \vartheta^0)/\varepsilon$.
4: Determine $A_{\alpha i}^w(\nu^0)$, $A_{\beta i}^w(\nu^0)$, $J_i^s(\nu^0)$ for $i = 1, \ldots, m$, $A_a^w(\nu^0)$, $A_b^w(\nu^0)$, $J_i^w(\nu^0)$ and $A_{a_1}^w(\nu^0)$, $A_{a_2}^w(\nu^0)$ and $J_2^w(\nu^0)$ from (1.85) and (1.86), respectively;
5: repeat
6: if snapflag = true then
7: Compute the solution $(y^{k+1}, p^{k+1})$ by solving (1.96);
8: if $k > K$ then
9: Set snapflag = false;
2.3 POD-based augmented Lagrangian method

In the augmented Lagrangian method, the evaluation of the gradient and of the step size can be costly, in particular in the projected-gradient algorithm. Therefore, it seems reasonable to try a reduced-order approach based on POD to approximate the state and adjoint solutions. According to Remark 2.3, the POD approximation of the state solution is $y^\ell = \hat{y} + S^\ell u$, where $S^\ell$ is defined in (2.7). The POD approximation of the dual equation, instead, is defined as $p^\ell = \hat{p} + A^\ell_1 u + A^\ell_4 (u, \mu_a, \mu_b)$ for $\mu_a, \mu_b \in W$, where for any $g \in L^2(0, T; \mathbb{R}^m)$ we have that $q = A^\ell_1 g$ satisfies

$$-\frac{d}{dt} \langle q(t), \psi \rangle_H + a(t; \psi, q(t)) = -\sigma_Q \langle (S^\ell g)(t), \psi \rangle_H, \quad q(T) = -\sigma_T (S^\ell g)(T)$$

10: Generate a POD-basis of rank $\ell$ according to $\tau_1$ using the computed $(y^{k+1}, p^{k+1})$ as snapshots;
11: end if
12: else
13: Compute the POD solution $(y^{k+1}, p^{k+1})$ according to (2.9)-(2.10);
14: Compute the a-posteriori error estimate $e$ from (2.14);
15: if $e > \tau_2$ then
16: Set snapflag = true;
17: end if
18: end if
19: Compute $\nu^{k+1} = (u^{k+1}, w^{k+1}, \vartheta^{k+1}) \in Z \times W$ from (1.89a)-(1.89i) and set $k = k + 1$;
20: Determine $A^\ell_{a1}(\nu^k), A^\ell_{b1}(\nu^k), J^\ell_{ai}(\nu^k)$ for $i = 1, \ldots, m, A^W_a(\nu^k), A^W_b(\nu^k), J^W_a(\nu^k)$ and $A^W_{a2}(\nu^k), A^W_{b2}(\nu^k)$ and $J^W_2(\nu^k)$ from (1.85) and (1.86), respectively;
21: if $A^\ell_{a1}(\nu^k) = A^\ell_{a1}(\nu^{k-1})$ and $A^\ell_{b1}(\nu^k) = A^\ell_{b1}(\nu^{k-1})$ for $i = 1, \ldots, m$ then
22: if $A^W_a(\nu^k) = A^W_a(\nu^{k-1})$ and $A^W_b(\nu^k) = A^W_b(\nu^{k-1})$ then
23: if $A^W_{a2}(\nu^k) = A^W_{a2}(\nu^{k-1})$ and $A^W_{b2}(\nu^k) = A^W_{b2}(\nu^{k-1})$ then
24: flag = true;
25: end if
26: end if
27: end if
28: until flag = true;
for all $ψ \in S^t$ a.e. in $[0,T)$. Moreover, for any $g \in L^2(0,T;\mathbb{R}^m)$ we have that $q = A^t_i(g,μ_a,μ_b)$ for $μ_a,μ_b \in W$ satisfies

$$-\frac{d}{dt} \langle q(t), ψ \rangle_H + a(t;ψ,q(t)) = c \left\langle \max \left\{ 0, \hat{y}_a(t) - (S^t g)(t) + \frac{μ_a(t)}{c} \right\}, ψ \right\rangle_H$$

$$- c \left\langle \max \left\{ 0, (S^t g)(t) - \hat{y}_b(t) + \frac{μ_b(t)}{c} \right\}, ψ \right\rangle_H,$$

$$q(T) = c \left\langle \max \left\{ 0, (S^t g)(T) - \hat{y}_a(T) + \frac{μ_a(T)}{c} \right\}, ψ \right\rangle_H$$

$$- c \left\langle \max \left\{ 0, (S^t g)(T) - \hat{y}_b(T) + \frac{μ_b(T)}{c} \right\}, ψ \right\rangle_H$$

for all $ψ \in S^t$ a.e. in $[0,T)$. The POD reduced gradient can be computed accordingly. As in Section 2.2, we get an improvement of the computational time, of course paying in accuracy of the approximation. Moreover, the same issues of Remark 2.4 arise for the augmented Lagrangian method. Since in this framework we are exclusively interested in the limit for the penalty parameter $c_k$, we propose to solve the full-order augmented Lagrangian problem ($\hat{L}_k$) for few steps $k = 0, \ldots, K$ in order to compute good snapshots. If $K$ is sufficiently small, one can still compute the POD solution of the ALM with a good speed-up. In fact, at the begin the projected gradient (or the projected BFGS) should not take many iterates to converge, since the penalty parameter $c_k$ is still sufficiently small to not strongly enforce the state constraints, which are the reason for an increasing computational challenge for the optimization solver. This strategy leads to a further issue: if the parameter $c_k$ gets too large, the state and adjoint snapshots do not have the same order of magnitude, therefore, when we solve the minimization problem (2.3) to compute the POD basis, the first POD bases would contain more (or even exclusively) information about the dynamics of the dual equation, causing problems in the POD approximation. As in [78], we fix this problem dividing the state and adjoint snapshots by different quantities, e.g., by their own variance. This strategy turns out to be successful in [78], but the choice of the scaling parameter is also heuristic and depends on the problem formulation. Anyway, we will use this technique in our numerical simulations. An a-posteriori error estimate for the POD-ALM is derived in Section 2.4.2. This estimate is used to ensure the quality of the POD approximation, especially when the parameter $c$ increases significantly, implying that the initial snapshots are no more representative of the dynamics of the full-order model. In this case, it is necessary to update the POD basis. In [78], this a-posteriori estimate is not utilized. Therefore, in addition
to [78], we consider a basis update strategy, cf. Algorithm 7.

**Algorithm 7** (POD-based augmented Lagrangian method)

1: **Data:** Initial pair \((\mu_{c_a}^0, \mu_{c_b}^0) \in \mathcal{W} \times \mathcal{W}\), initial weight \(c_0 > 0\), increment \(\beta > 0\) for \(c_n\), tolerance \(\varepsilon > 0\), maximum number of iterations \(n_{max}\), residuals \(R^{0, +}_{c_a} \gg 1\), \(R^{0, +}_{c_b} \gg 1\), residual weight \(\tau \in (0, 1)\), number of iteration to generate the snapshots \(K\), tolerance \(\tau_1\) for the a-priori estimate (2.11) and tolerance \(\tau_2\) for the a-posteriori error estimate (2.30);

2: set \(n = 0, k = 0\), \(\text{flag} = \text{false}\) and \(\text{snapflag} = \text{true}\);
3: while \(\text{flag} \) and \(n < n_{max}\) do
4: if \(\text{snapflag} = \text{true}\) then
5: Follow Step 4 - Step 20 of Algorithm 3;
6: if \(n \geq 1\) and \(k > K\) then
7: Set \(\text{snapflag} = \text{false}\);
8: Scale the snapshots \(y^n = Su^n\) and \(p^n = A_1u^n + A_4(u^n, \mu_{c_a}^k, \mu_{c_b}^k)\), e.g., with their own variance;
9: Generate a POD-basis of rank \(\ell\) according to \(\tau_1\) using the scaled \((y^n, p^n)\) as snapshots;
10: end if
11: else
12: For fixed \((\mu_{c_a}^k, \mu_{c_b}^k)\) find \(u^{k+1}\) solving the problem

\[
\min \hat{L}^\ell_{c_k}(u, \mu_{c_a}^k, \mu_{c_b}^k) \text{ s.t. } u \in \mathcal{U}_{ad}; \quad (\hat{L}^\ell_k)
\]

13: Follow Step 5 - Step 20 of Algorithm 3;
14: Compute the a-posteriori error estimate \(e\) from (2.30);
15: if \(e > \tau_2\) then
16: Set \(\text{snapflag} = \text{true}\);
17: end if
18: end if
19: end while

**Remark 2.5.** Note that in Step 8 we do not need to include \(\hat{y}\) and \(\hat{p}\) in the snapshots, because they are independent from the control \(u\) and they can be precomputed once for all the steps in the ALM algorithm. In fact, thanks to the linearity of the state and adjoint equations, we have \(y^\ell = \hat{y} + S^\ell u\) and \(p^\ell = \hat{p} + A_1^\ell u + A_4^\ell(u, \mu_{c_a}^k, \mu_{c_b}^k)\).
Moreover, the condition \( n \geq 1 \) in Step 6 is to ensure that at least there was one successful step before computing the POD basis, otherwise we know a-priorily that the snapshots can not be a good approximation of the optimal solution.

\[ \diamond \]

2.4 A-posteriori error estimates

As previously mentioned, the approximation error in the reduced-order model depends on the snapshots. The better the snapshots represent the dynamics involved in the problem the more the reduced-order model is accurate. Furthermore, the rank of the POD basis \( \ell \) is also playing a role in the approximation quality: the greater \( \ell \) is the better the model can reconstruct the dynamics. Therefore, two questions arise naturally:

- How large does one have to choose \( \ell \)?
- How is it possible to measure the error between the reduced-order and the full-order solutions without computing the full-order one?

To answer these two questions an a-posteriori error analysis can be used together with the a-priori error \( (2.6) \). In this section we derive an a-posteriori error analysis for the virtual control approach and the augmented Lagrangian method.

2.4.1 A-posteriori error estimate for the virtual control approach

In this section we present an a-posteriori error estimate which is based on a perturbation argument [33] and has been already utilized in [127]. As done in [43], this estimate can be generalized for the mixed control-state constraints case. In addition to [43], the virtual control \( w \) is also constrained, causing additional restrictions to the estimate; see Remark 2.7. At first, the first-order sufficient optimality conditions for \((\hat{P})\) are given as

\[ \langle \nabla \hat{J}(\bar{z}), z - \bar{z} \rangle_Z \geq 0 \quad \text{for all } z \in Z_{ad}, \quad (2.12) \]

where the gradient \( \nabla \hat{J} \) of \( \hat{J} \) at a given \( z = (u, w) \in Z_{ad} \) has been given in \((1.38)\). Note that we do not indicate explicitly the dependence of the solution on \( \varepsilon \), in order to simplify the notation. Moreover, we recall that the linear, invertible
operator $\mathcal{T}$ has been introduced in (1.26). Notice that $z = (u, w)$ belongs to $\mathcal{Z}_{ad}$ if $\bar{z} = (u, w) = \mathcal{T}(z) \in \mathcal{Z}_{ad} \subseteq \mathcal{Z}$ holds with the closed, bounded and convex subset

$$3_{ad} = \{(u, w) \in \mathcal{Z} \mid u_a \leq u \leq u_b, \bar{y}_a \leq w \leq \bar{y}_b \text{ and } w_a \leq \frac{1}{\varepsilon}(w - Su) \leq w_b\}.$$ 

Instead of $(\hat{\mathcal{P}})$, we consider the following optimal control problem

$$\min \tilde{J}(T^{-1}\bar{z}) \quad \text{s.t.} \quad \bar{z} = (u, w) \in 3_{ad}.$$ 

If $\bar{z} = (\bar{u}, \bar{w})$ solves $(\hat{\mathcal{P}})$, then $\bar{z} = \mathcal{T}(\bar{z})$ is the solution to $(\hat{\mathcal{P}})$. Conversely, if $\bar{z}$ solves $(\hat{\mathcal{P}})$, then $\bar{z} = T^{-1}(\bar{z})$ is the solution to $(\hat{\mathcal{P}})$. First-order sufficient optimality conditions for $(\hat{\mathcal{P}})$ are

$$\langle T^{-*} \nabla \tilde{J}(T^{-1}\bar{z}), \bar{z} - \bar{z} \rangle_z \geq 0 \quad \text{for all } \bar{z} = (u, w) \in 3_{ad}, \quad (2.13)$$

where $T^{-*} : \mathcal{Z} \rightarrow \mathcal{Z}$ denotes the adjoint of the operator $T^{-1}$. Now, we can state the main result of this section in the following theorem:

**Theorem 2.6.** Let Assumptions 1.1 and 1.12 hold true. Let $\bar{z} = (\bar{u}, \bar{w})$ be the optimal solution to $(\hat{\mathcal{P}})$. Then, $\bar{z} = \mathcal{T}(\bar{z})$ is the solution to $(\hat{\mathcal{P}})$. Suppose that $z^{ap} = (u^{ap}, w^{ap}) \in \mathcal{Z}_{ad}$ is an approximate solution to $(\hat{\mathcal{P}})$. We set $\bar{z}^{ap} = \mathcal{T}(z^{ap})$. Then, there exists a perturbation $\bar{\zeta} = (\zeta^u, \zeta^w) \in \mathcal{Z}$, which is independent of $\bar{z}$, so that

$$\|\bar{z} - z^{ap}\|_z \leq \frac{1}{\sigma_z} \|T^* \bar{\zeta}\|_z \quad \text{with } \sigma_z = \min\{\sigma_1, \ldots, \sigma_m, \sigma_w\} > 0,$$

where $T^*$ denotes the adjoint of the operator $T$; cf. (1.28).

**Proof.** Since $T$ has a bounded inverse, the first part follows. Since $z^{ap} = (u^{ap}, w^{ap}) \in \mathcal{Z}_{ad}$, then $z^{ap} = \mathcal{T}(z^{ap}) \in 3_{ad}$. Therefore, recalling (2.13), there exists a perturbation $\zeta = (\zeta^u, \zeta^w) \in \mathcal{Z}$ so that

$$\langle T^{-*} \nabla \tilde{J}(z^{ap}) + \zeta, \bar{z} - z^{ap} \rangle_z \geq 0 \quad (2.15)$$

for all $\bar{z} = (u, w) \in 3_{ad}$; cf. [9, 93]. Let $\tilde{p} = \hat{p} + \mathcal{A} \tilde{u}$ and $p^{ap} = \hat{p} + \mathcal{A} u^{ap}$. Moreover, we set $y^{ap} = \tilde{y} + \mathcal{S} u^{ap}$ and we have $\tilde{y} = \tilde{y} + \mathcal{S} \tilde{u}$. Then, $\tilde{p}$ and $p^{ap}$ satisfy

$$\int_0^T -\langle \tilde{p}(t), \varphi(t) \rangle_{V', V} + a(t; \tilde{y}(t), \tilde{p}(t)) \, dt$$

$$\quad = \int_0^T \sigma_Q \langle (y_Q - \tilde{y})(t), \varphi(t) \rangle_H \, dt \quad \text{for all } \varphi \in W(0, T),$$

$$\tilde{p}(T) = \sigma_T (y_T - \tilde{y}(T)) \quad \text{in } H \quad (2.16a)$$

$$\tilde{p}(T) = \sigma_T (y_T - \tilde{y}(T)) \quad \text{in } H \quad (2.16b)$$
and
\[
\int_0^T -\langle \ddot{y}^{\text{ap}}(t), \varphi(t) \rangle_{V', V} + a(t; \varphi(t), y^{\text{ap}}(t)) \, dt = \int_0^T \sigma_Q \langle (y_Q - y^{\text{ap}})(t), \varphi(t) \rangle_H \, dt \quad \text{for all } \varphi \in W(0, T),
\]
\[
p^{\text{ap}}(T) = \sigma_T (y_T - y^{\text{ap}}(T)) \quad \text{in } H, \tag{2.17b}
\]
respectively. Recall the adjoint operator \( B^*: L^2(0, T; V) \to \mathcal{U} \) from (1.2). Choosing \( \bar{z} = \bar{z}^{\text{ap}} \) in (2.13), \( \bar{z} = \bar{z} \) in (2.15) and adding both inequalities we infer that
\[
0 \leq \left\langle T^*(\nabla \dot{J} (T^{-1} \bar{z}^{\text{ap}}) + T^* \zeta - \nabla \dot{J} (T^{-1} \bar{z})), \bar{z} - \bar{z}^{\text{ap}} \right\rangle_Z
\]
\[
= \left\langle \nabla \dot{J}(\bar{z}^{\text{ap}}) - \nabla \dot{J}(\bar{z}) + T^* \zeta, T^{-1}(\bar{z} - \bar{z}^{\text{ap}}) \right\rangle_Z
\]
\[
= \left\langle \sum_{1 \leq i \leq m} (\sigma_i (u_i^{\text{ap}} - \bar{u}_i) - \gamma_c \langle b_i, p^{\text{ap}}(\cdot) - \bar{p}(\cdot) \rangle_{L^2(\Gamma_i)})_{1 \leq i \leq m}, \bar{z} - \bar{z}^{\text{ap}} \right\rangle_Z
\]
\[
+ \left\langle T^* \zeta, \bar{z} - \bar{z}^{\text{ap}} \right\rangle_Z
\]
\[
\leq -\sigma_z \| \bar{z} - \bar{z}^{\text{ap}} \|_Z^2 - \langle B(\bar{u} - u^{\text{ap}}), p^{\text{ap}} - \bar{p} \rangle_{L^2(0, T; V'}, L^2(0, T; V)}
\]
Using integration by parts and (2.16a), (2.17a) we have
\[
\langle B(\bar{u} - u^{\text{ap}}), p^{\text{ap}} - \bar{p} \rangle_{L^2(0, T; V'), L^2(0, T; V)}
\]
\[
= \int_0^T \gamma_c \sum_{i=1}^m (\bar{u}_i - u_i^{\text{ap}})(t)(p^{\text{ap}} - \bar{p})(t) \, dt
\]
\[
= \int_0^T \langle (\bar{y} - y^{\text{ap}})(t), (p^{\text{ap}} - \bar{p})(t) \rangle_{V', V} + a(t; (\bar{y}(t) - y^{\text{ap}})(t), (p^{\text{ap}} - \bar{p})(t)) \, dt
\]
\[
= \int_0^T -\langle (p^{\text{ap}} - \bar{p})(t), (\bar{y} - y^{\text{ap}})(t) \rangle_{V', V} + a(t; (\bar{y}(t) - y^{\text{ap}})(t), (p^{\text{ap}} - \bar{p})(t)) \, dt
\]
\[
+ \langle (\bar{y} - y^{\text{ap}})(T), (p^{\text{ap}} - \bar{p})(T) \rangle_H - \langle (\bar{y} - y^{\text{ap}})(0), (p^{\text{ap}} - \bar{p})(0) \rangle_H.
\]
We have \( \bar{y}(0) = y^{\text{ap}}(0) = y_0 \). Moreover, (2.16b) and (2.17b) yield
\[
(p^{\text{ap}} - \bar{p})(T) = \sigma_T (\bar{y}(T) - y^{\text{ap}}(T)).
\]
Thus, utilizing (2.16a) and (2.17a) with \( \varphi = \bar{y} - y^{ap} \) we get
\[
\langle B(\bar{u} - u^{ap}), p^{ap} - \bar{p} \rangle_{L^2(0,T;V'), L^2(0,T;V)} \\
= \int_0^T -\langle p^{ap}(t), (\bar{y} - y^{ap})(t) \rangle_{V', V} + a(t; (\bar{y}(t) - y^{ap})(t), p^{ap}(t)) \, dt \\
- \int_0^T -\langle \bar{p}(t), (\bar{y} - y^{ap})(t) \rangle_{V', V} + a(t; (\bar{y}(t) - y^{ap})(t), \bar{p}(t)) \, dt \\
+ \sigma_T \| (\bar{y} - y^{ap})(T) \|_H^2 \\
= \sigma_Q \| \bar{y} - y^{ap} \|_{L^2(0,T;H)}^2 + \sigma_T \| (\bar{y} - y^{ap})(T) \|_H^2.
\]
Combining (2.18) and (2.19) we obtain
\[
\sigma_z \| \bar{z} - z^{ap} \|_Z^2 \leq -\sigma_Q \| \bar{y} - y^{ap} \|_{L^2(0,T;H)}^2 - \sigma_T \| (\bar{y} - y^{ap})(T) \|_H^2 + \langle T^* \zeta, \bar{z} - z^{ap} \rangle_Z
\]
which implies that
\[
\| \bar{z} - z^{ap} \|_Z \leq \frac{1}{\sigma_z} \| T^* \zeta \|_Z.
\] (2.20)

The perturbation \( \zeta \) can be computed as follows: Let \( \xi = (\xi^u, \xi^w) \in Z \) be given as \( \xi = T^* \nabla \hat{J}(z^{ap}) \in Z \). Then, \( \xi \) solves the linear system \( T^* \zeta = \nabla \hat{J}(z^{ap}) \), i.e.,
\[
\begin{pmatrix}
\mathcal{I}_d & S^* \\
\mathcal{E} \mathcal{I}_W & 0
\end{pmatrix}
\begin{pmatrix}
\xi^u \\
\xi^w
\end{pmatrix} = \begin{pmatrix}
\left( \frac{\sigma_i u^{ap}_i - \gamma_c \int_{\Gamma_c} b_i p^{ap} \, ds}{\sigma_u w^{ap}} \right)_{1 \leq i \leq m}
\end{pmatrix}
\]
(2.21)
where \( p^{ap} = \hat{p} + A_1 u^{ap} \). Note that (2.15) can be written as
\[
\langle \xi + \zeta, \bar{z} - z^{ap} \rangle_Z \geq 0 \quad \text{for all } \bar{z} \in \mathfrak{z}_{ad}.
\]
Thus, to ensure the previous inequality, we have that
\[
\zeta_i^u(t) = \begin{cases}
-\min\{0, \xi_i^u(t)\} & \text{for } t \in A_i^{(l)}(z^{ap}), \\
-\max\{0, \xi_i^u(t)\} & \text{for } t \in A_i^{(l)}(z^{ap}), \\
-\xi_i^u(t) & \text{for } t \in J_i^{(l)}(z^{ap})
\end{cases}
\] (2.22a)
for \( i = 1, \ldots, m \) and
\[
\zeta^w(t, x) = \begin{cases}
-\min\{0, \xi^w(t, x)\} & \text{for } (t, x) \in A_i^{(w)}(z^{ap}), \\
-\max\{0, \xi^w(t, x)\} & \text{for } (t, x) \in A_i^{(w)}(z^{ap}), \\
-\xi^w(t, x) & \text{for } (t, x) \in J_i^{(w)}(z^{ap})
\end{cases}
\] (2.22b)
Hence, $\zeta$ is independent from $J^w_2(z^{ap})$, $A^W_{a,2}(z^{ap})$ and $A^W_{b,2}(z^{ap})$, but the constraints $w_a$ and $w_b$ impose restrictions on the perturbation $z^{ap}$, as will be shown in Remark 2.7. Note that we can easily decouple the equations in system (2.21) by computing $\xi^w = \varepsilon^{-1} \sigma_w u^{ap}$ from the second equation and plugging it in the first one in order to have

$$\xi^w = -S^* \xi^w + \left( \sigma_i u_i^{ap} - \gamma_c \int_{\Gamma_e} b_i p^{ap} \, ds \right)_{1 \leq i \leq m}$$

$$= -\frac{\sigma_w}{\varepsilon} S^* u^{ap} + \left( \sigma_i u_i^{ap} - \gamma_c \int_{\Gamma_e} b_i p^{ap} \, ds \right)_{1 \leq i \leq m}$$

We recall that $S^*: W_0(0, T') \to \mathcal{U}$ denotes the dual operator of the linear solution operator $S$; cf. Remark 1.23. Moreover, the omission of the operator $E$ in $T$ has to be taken in account here; see Remark 1.22. Following the proof of Lemma 2.4 in [127], we obtain

$$(S^* E^* \xi^w)(t) = B^* p^w(t)$$

for all $t \in [0, T]$, where $p^w$ is the solution of the dual equation (c.f. (1.30)) with $y_Q = \xi^w$, $y_T = \xi^w(T)$, $\sigma_Q = \sigma_T = 1$, $y = 0$ and $\beta = 0$.

**Remark 2.7.** In our numerical realization the approximate solution $z^{ap}$ is given by the POD suboptimal solution $\bar{z} = (\bar{u}^\ell, \bar{w}^\ell) \in Z^\ell_{ad}$ to the POD Galerkin scheme for (1.96). Thus, (2.14) can be utilized as an a-posteriori error estimate. Due to the PDASS algorithm structure, the suboptimal POD solution satisfies the constraints

$$u_a \leq \bar{u}^\ell \leq u_b, \quad y_a \leq \varepsilon \bar{w}^\ell + \bar{y}^\ell \leq y_b, \quad w_a \leq \bar{w}^\ell \leq w_b$$

with $\bar{y}^\ell = \dot{\bar{y}}^\ell + S^f \bar{u}^\ell$. Therefore, we set

$$z^{ap} = (u^{ap}, w^{ap}) \in \mathcal{Z} \quad \text{with} \quad u^{ap} = \bar{u}^\ell \quad \text{and} \quad w^{ap} = \varepsilon \bar{w}^\ell + \bar{y}^\ell + S^f \bar{u}^\ell - \dot{\bar{y}}.$$  \hfill (2.23)

To apply the a-posteriori estimate, $z^{ap}$ has to be in $\mathcal{Z}_{ad}$. In particular, we need that

$$w_a \leq \varepsilon^{-1}(w^{ap} - Su^{ap}) \leq w_b,$$

i.e., that

$$w_a \leq \bar{w}^\ell + \frac{1}{\varepsilon} (\bar{y}^\ell + S^f \bar{u}^\ell - \dot{\bar{y}} - S\bar{u}^\ell) \leq w_b.$$  \hfill (2.24)

In general, this condition may not be satisfied, therefore it has to be checked, before applying the a-posteriori error estimate. In our application, this condition is not restrictive, since we can choose the artificial bounds $w_a$ and $w_b$ very small and very
large, respectively. Therefore, for large enough $\varepsilon$ and $\ell$ the term $\varepsilon^{-1}(\hat{y}^\ell + S^\ell \hat{u}^\ell - \hat{y} - S\bar{u}^\ell)$ should be small enough to guarantee (2.24). Throughout the thesis we assume that (2.24) is verified. From $\bar{z}^\ell \in Z_{ad}$ and from the previous assumption, we infer that $\bar{z}^\ell \in Z_{ad}$. It follows from (1.27) and (2.23) that

$$z^{ap} = T^{-1}(\bar{z}^\ell) = \left( w^{ap}, \varepsilon^{-1}(w^{ap} - Su^{ap}) \right)$$

fulfills (2.14). Moreover, we find that

$$\bar{z} - z^{ap} = \bar{z} - \bar{z}^\ell + (0, \varepsilon^{-1}(\bar{y}^\ell - \bar{y}^\ell)).$$

(2.25)

where $\bar{y}^\ell = \bar{y} + S\bar{u}^\ell$ is the solution of (1.3) for $u = \bar{u}^\ell$. If $\zeta = (\zeta^u, \zeta^w)$ is computed according to (2.22) we have that

$$\|\bar{u} - \tilde{u}^\ell\|_U^2 \leq \|\bar{z} - z^{ap}\|_Z^2 \leq \frac{1}{\sigma_z} \|T^*\zeta\|_Z.$$

(2.26)

Therefore, we have to evaluate

$$T^*\zeta = \begin{pmatrix} I_u & S^* \\ 0 & \varepsilon I_w \end{pmatrix} \begin{pmatrix} \zeta^u \\ \zeta^w \end{pmatrix} = \begin{pmatrix} \zeta^u + S^*\zeta^w \\ \varepsilon\zeta^w \end{pmatrix},$$

in order to get an a-posteriori error estimate for $\|\bar{u} - \tilde{u}^\ell\|_U^2$. ♦

2.4.2 A-posteriori error estimate for the augmented Lagrangian method

Let $u^{ap} \in U_{ad}$ be an approximation of the optimal solution $\bar{u}$ of $(\hat{L})$ for fixed $c > 0$ and $\mu_a, \mu_b \in W$. Using again the perturbation argument in [9, 33, 93], we have that there exists a perturbation $\zeta$ such that

$$\langle \nabla \hat{L}_c(u^{ap}, \mu_a, \mu_b) + \zeta, u - u^{ap} \rangle_U \geq 0 \text{ for all } u \in U_{ad}. \quad (2.27)$$

Moreover, the optimal control $\bar{u}$ to $(\hat{L})$ satisfies the following variational inequality

$$\langle \nabla \hat{L}_c(\bar{u}, \mu_a, \mu_b), u - \bar{u} \rangle_U \geq 0 \text{ for all } u \in U_{ad}. \quad (2.28)$$
Choosing \( u = u^{ap} \) in (2.28) and \( u = \bar{u} \) in (2.27), we obtain from (1.104)

\[
0 \leq \langle \nabla \hat{L}_c(u^{ap}, \mu_a, \mu_b), \bar{u} - u^{ap} \rangle_u \\
= \sum_{i=1}^{m} (\sigma_i u_i^{ap} - \gamma_c \langle b_i, p^{ap} \rangle_{L^2(\Gamma_c)} + \zeta_i - \sigma_i \bar{u}_i + \gamma_c \langle b_i, \bar{p} \rangle_{L^2(\Gamma_c)}, \bar{u}_i - u_i^{ap})_{L^2(0,T)} \\
\leq -\sigma_u \| \bar{u} - u^{ap} \|^2_u + \| \zeta \|_u \| \bar{u} - u^{ap} \|_u - \int_0^T \sum_{i=1}^{m} \gamma_c (\bar{u}_i - u_i^{ap}) \langle b_i, p^{ap}(t) - \bar{p}(t) \rangle_{L^2(\Gamma_c)} dt,
\]

where \( p^{ap} = \hat{p} + A_1 u^{ap} + A_4(u^{ap}, \mu_a, \mu_b), \bar{p} = \hat{p} + A_1 \bar{u} + A_4(\bar{u}, \mu_a, \mu_b) \) and \( \sigma_u = \min \{\sigma_1, \ldots, \sigma_m\} \). Now, setting \( u^\delta = \bar{u} - u^{ap} \) in Lemma 1.50, we obtain

\[
\int_0^T \sum_{i=1}^{m} \gamma_c (\bar{u}_i - u_i^{ap}) \langle b_i, p^{ap}(t) - \bar{p}(t) \rangle_{L^2(\Gamma_c)} dt = \sigma_Q \| S(\bar{u} - u^{ap}) \|^2_{L^2(0,T;H)} + \sigma_T \| S(\bar{u} - u^{ap})(T) \|^2_H
\]

Hence, the following estimate

\[
\| \bar{u} - u^{ap} \|_u \leq \frac{1}{\sigma_u} \| \zeta \|_u \tag{2.30}
\]

holds true. The perturbation \( \zeta \) can be computed as follows: Let \( \xi \in U \) be given as \( \xi = \nabla \hat{L}_c(u^{ap}, \mu_a, \mu_b) \). Note that (2.27) can be written as

\[
\langle \xi + \zeta, u - u^{ap} \rangle_u \geq 0 \text{ for all } u \in U_{ad}
\]

Thus, to ensure the previous inequality, we have to choose \( \zeta \in U \) as follows

\[
\zeta_i(t) = \begin{cases} 
-\min \{0, \xi_i(t)\} & \text{for } u_i^{ap} = u_{ai}, \\
-\max \{0, \xi_i(t)\} & \text{for } u_i^{ap} = u_{bi}, \\
-\xi_i(t) & \text{otherwise}
\end{cases}
\]

for almost all \( t \in [0, T] \) and \( i = 1, \ldots, m \).

**Remark 2.8.** Analogously to Section 2.4.1, in our numerical realization the approximate solution \( u^{ap} \) is given by the POD suboptimal solution \( \bar{u}^\ell \) to the POD Galerkin scheme for \( \hat{L} \). Thus, (2.30) can be utilized as an a-posteriori error estimate, if \( \zeta \) is computed according to (2.31). \( \diamond \)
2.5 Numerical tests

For these numerical tests we refer to the test data in Section 1.5. Moreover, we set the parameters of the optimal control problem as in Section 1.5.2. Let us point out that also here the choice of an economic optimal control problem, i.e., \( \sigma_T = \sigma_Q = 0 \), does not change the proofs of the estimates in Section 2.4. We apply the POD method introduced in this chapter, studying especially the effectiveness of the a-posteriori error estimators. These estimators are important to certify the validity of the POD approximation without computing the full-order model solution, which is in general really costly in terms of computational time. As it is well-known, the quality of the POD approximation depends on the snapshots and the number of POD basis functions; cf. [57, 68, 85]. In [95], we study how the POD-PDASS approximation error behaves for different snapshots choices, including the best possible one, i.e., the optimal solution, that is clearly unknown without solving the optimal control problem with the full-order model, making its use impossible. A similar study for the ALM-POD algorithm can be found in [78]. In this thesis we do not repeat this kind of tests, which are also well commented in the literature, e.g. in [57, 61, 78, 85, 95] and many others. Here, with our application in mind, we focus on different aspects of the POD method. In Section 2.5.1, we study the behaviour of the POD reduced-order model and of the a-posteriori error estimator (2.14) for different values of \( \varepsilon \). In Section 2.5.2, instead, we propose a different way to generate the POD snapshots compared to the ones described in [78], making use of the a-posteriori error estimator (2.30). This new strategy justifies a later approach for the MPC-POD snapshots generation in Chapter 3.

2.5.1 PDASS a-posteriori error estimator

For this test we generate the snapshots with the optimal state and adjoint solutions, in such a way that the true error, i.e., the norm of the difference between the optimal FE control and the reduced-order optimal one, is going to zero when the number of POD basis functions \( \ell \) is increasing. As observed in Remark 2.7, we need that (2.24) hold. In all of the tests in this section, we checked this assumption and it is verified for all the choices of \( \varepsilon \) and \( \ell \) we make. As one can see from Figure 2.1, the eigenvalues corresponding to the POD basis functions decay faster for greater values of \( \varepsilon \). Therefore, taking into account (2.6), for larger \( \varepsilon \) we need less basis functions to reconstruct the snapshots, i.e., the optimal solution. This is because the optimal solution violates less the constraints for a decreasing \( \varepsilon \), as shown in Section 1.5.2,
and thus the optimal state and adjoint trajectories differ much more. This is also confirmed by the results reported in Table 2.1. To simplify the notation, we set $A = A_a^W(\bar{\nu}) \cup A_b^W(\bar{\nu})$ and $A^\ell = A_a^W(\bar{\nu}^\ell) \cup A_b^W(\bar{\nu}^\ell)$. The relative errors
\[ \text{rel_err}_u = \frac{||\bar{u} - \bar{u}^\ell||_u}{||\bar{u}||_u}, \quad \text{rel_err}_\text{AS} = \frac{|A \cup A^\ell \setminus A \cap A^\ell|}{|A \cup A^\ell|}, \]
measure the difference between the FE and POD optimal solutions in terms of optimal controls and of points belonging to the space-time grid in which the solutions violate the constraints. For the same number of POD basis functions $\ell$, these errors are smaller for $\varepsilon = 0.5$ and increase for decreasing $\varepsilon$, as expected. Instead, the averaged computational time speed-up to solve one PDASS iteration has the same order of magnitude, due to the fact that $\ell$ is fixed and therefore also the discrete linear system dimension is the same for all $\varepsilon$. Similarly, the time speed-up for the overall algorithm is of the same order for all values of $\varepsilon$, but it is significantly reduced compared to the one for the averaged time. This is due to the fact that we can not avoid the computation of the active sets. This computation can only be done on
the FE level, because we do not have the knowledge of the pointwise value of the solution in the reduced-order space. Another costly part is the assembling of the POD-PDASS iteration matrix, since each reduced matrix is dense. Therefore, the assembling time is almost the same, despite the fact that the reduced-order linear system is smaller than the full-order sparse one. This last issue can be bypassed using a matrix-free implementation of the POD-PDASS algorithm. In Figure 2.2,

![Graph](image)

Figure 2.2: Comparison between true and a-posteriori errors for different $\varepsilon$, plotted in semi-logarithmic scale.

we plot the true and the estimated a-posteriori errors for different values of $\varepsilon$. By true error we mean $\|\bar{u} - \bar{u}^\ell\|_U$. As one can notice, the gap between the a-posteriori estimate and the true error increases as soon as $\varepsilon$ decreases. The reason is hidden
in the estimate (2.14), where to estimate the true error we get rid of the term
\[ \| \bar{w} - w^{ap} \|_{W}^2 = \| \bar{w} - \bar{w}^{\ell} + \varepsilon^{-1} (\bar{y}^{\ell} - \tilde{y}^{\ell}) \|_{W}^2 \] (2.32)
with \( \tilde{y}^{\ell} = \tilde{y} + S\bar{u}^{\ell} \) and \( \bar{y}^{\ell} = \bar{y}^{\ell} + S\bar{u}^{\ell} \). The term in (2.32) gets larger as \( \varepsilon \) decreases, making the a-posteriori estimate less accurate. As one can notice from Figure 2.2, this overestimation is then balanced by the fact that the differences \( \tilde{y}^{\ell} - \bar{y}^{\ell} \) and \( \bar{w} - \bar{w}^{\ell} \) get small as soon as \( \ell \) increases. In fact, the gap between true and estimated error reduces progressively for increasing \( \ell \). The tightness of the error estimator is crucial to certify the validity of the POD approximation and decide when it is necessary to update the POD basis, i.e., when to compute new snapshots; cf. Section 3.3.

### 2.5.2 POD-based ALM and a-posteriori error estimator

In [78], we discussed some possibilities to generate the POD snapshots for the augmented Lagrangian problem. We showed also how scaling the snapshots by their variance improves the POD approximation. According to [78], the best snapshots are, of course, the optimal ones. In this section, we compare the results obtained using Algorithm 7 and Algorithm 8.

**Algorithm 8** (POD-based augmented Lagrangian method with optimal snapshots)

1. **Data:** Initial pair \((\mu_c^0, \mu_b^0) \in W \times W\), initial weight \(c_0 > 0\), increment \(\beta > 0\) for \(c_n\), tolerance \(\varepsilon > 0\), \(n_{\text{max}}\) maximum number of iterations, \(R_0^{a,+} \gg 1\), \(R_0^{b,+} \gg 1\) residuals, residual weight \(\tau \in (0,1)\), tolerance \(\tau_1\) for the a-priori estimate (2.11) and optimal solutions \(\bar{y}\) and \(\bar{p}\);
2. set \(n = 0, k = 0, \text{flag} = \text{true}\);
3. Compute a POD-basis of rank \(\ell\) according to \(\tau_1\) using as snapshots the scaled \((\bar{y}, \bar{p})\);
4. **while** \(\text{flag} \) and \(n < n_{\text{max}}\) **do**
5. for fixed \((\mu_a^k, \mu_b^k)\) find \(a^{k+1}\) solving the problem \((\hat{L}_k^\ell)\);
6. Follow Step 5 - Step 20 of Algorithm 3;
7. **end while**

The first algorithm makes use of the a-posteriori error estimate (2.30) to decide when it is necessary to update the POD basis. In the second one, instead, there is no need to update the basis, since we know a-priorily that the optimal state and adjoint
solutions are the best snapshots we can use. Unfortunately, as already pointed out, Algorithm 8 can only be used for comparison purposes, because such an algorithm is not applicable in practice. According to Table 2.2, Algorithm 7 can be used as a valid alternative for Algorithm 8. Although the speed-up is way smaller, the relative error and the successful steps \( n \) are close. The increase of the computational time for Algorithm 7 is caused by two factors: the computation of the a-posteriori error estimate and the computation of new snapshots. Moreover, we choose \( K = 3 \) in Algorithm 7, which implies that we solve the optimal control problem with FE in the first three iterations. However, since \( c_k \) is small for \( k \leq K \), the inner iterations are not so many, making this approach reasonable. When there is a POD basis update, we solve the full-order optimal control problem (\( \hat{L}_k \)). This costly procedure is unfortunately necessary to keep a good approximation and it is justified by the fact that the optimal solutions for \( c_k \) and \( c_{k+1} \) should not be too different for a small enough increment \( \beta \). Compared to the techniques presented in [78], e.g., the random control approach, Algorithm 7 has the advantage of using the a-posteriori error estimate to check the quality of the approximation, thus it is more robust. As already pointed out, it is really important to have a robust technique to build the reduced-order model. For this reason, in Section 3.3 we use an a-posteriori update approach for the MPC-POD-PDASS algorithm; cf. Section 3.2. According to (2.30), we do not expect that the tightness of the a-posteriori error estimate for the ALM-POD is affected by the magnitude of the penalty parameter \( \epsilon \), as it happens instead for the estimate for POD-PDASS when \( \epsilon \) changes. This is confirmed by Figure 2.3, where the gap between the true error and the a-posteriori estimate is almost the same for different values of \( \epsilon \). To compute the plotted data, we choose an initial pair \((\mu^a, \mu^b) \in \mathcal{W} \times \mathcal{W}\) and we solve the full-order optimal control problem (\( \hat{L}_k \)) for a fixed \( \epsilon \). Then, we use the optimal state and adjoint solutions to generate a POD basis and we solve (\( \hat{L}_\ell \)) for \( \ell = 1, \ldots, \) the same \( \epsilon \) and the same \((\mu^a, \mu^b) \). Since we use the optimal snapshots, we expect that the true error \( \|\bar{u} - \bar{u}_\ell\|_U \) goes to zero.

<table>
<thead>
<tr>
<th>Method</th>
<th>rel_err_u</th>
<th>( k )</th>
<th>( n )</th>
<th>avg. it</th>
<th>Time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALM-BFGS-FE</td>
<td>–</td>
<td>63</td>
<td>40</td>
<td>162.3</td>
<td>7.10 h</td>
<td>–</td>
</tr>
<tr>
<td>ALM-BFGS-POD-Alg7</td>
<td>0.045</td>
<td>66</td>
<td>41</td>
<td>174.2</td>
<td>2.22 h</td>
<td>3.2</td>
</tr>
<tr>
<td>ALM-BFGS-POD-Alg8</td>
<td>0.021</td>
<td>64</td>
<td>40</td>
<td>164.7</td>
<td>0.52 h</td>
<td>13.6</td>
</tr>
</tbody>
</table>

Table 2.2: Comparison between ALM-BFGS-FE and ALM-BFGS-POD using Algorithm 7 and Algorithm 8 with a-priori tolerance \( \tau_1 = 10^{-5} \sum_{i=1}^{D} \lambda_i \).
Figure 2.3: True and a-posteriori errors (2.30) for different $c$, plotted in semi-logarithmic scale.

as $\ell$ increases. The oscillations occurring in Figure 2.3 for a large number of basis functions $\ell$ are caused by the tolerances set, e.g., the stopping criteria of the PG and of the Armijo rule. In fact, these oscillations occur when the true and a-posteriori errors are below $10^{-6}$. The a-posteriori estimate is affected by this phenomena as well, therefore it happens that for some indices $\ell$ the a-posteriori estimator is even below the true error.
Chapter 3

POD-based Model Predictive Control for linear-quadratic OCP

In this chapter, we introduce the Model Predictive Control (MPC) algorithm. This method is used mainly for infinite (or large) time horizon optimal control problems, which, obviously, can not be directly solved on the whole time interval. Here, we use this technique to avoid the curse of dimensionality in time for the PDASS iteration matrix; cf. Remark 1.48. Moreover, as will be shown in Section 3.1, MPC computes a feedback control and, thus, it is possible to update the parameters in the optimal control problem without destroying the convergence of the algorithm. This updating strategy is justified by our application, where it is not possible, for example, to have an exact long-term weather forecast for the outside temperature. Thus, it seems reasonable to progressively update it as long as the algorithm runs. Furthermore, in Section 3.2 we show how POD and MPC can be successfully combined to speed-up the computational time and preserving a good quality of the approximation of the solution.

3.1 Model Predictive Control

The basic idea of MPC is to predict, stabilize and optimize a given dynamical system by reconstructing the optimal control $u(t) = \Phi(t, y(t))$ in a feedback form. In order to do that, one has to solve repetitively many open-loop optimal control problems on smaller time horizons $N \Delta T \ll T$, $N \in \mathbb{N}$. Then, the open-loop control on the first time step is stored and used to compute the dynamical system trajectory, before solving the next open-loop problem on a shifted time interval. More details
on MPC and its properties can be found, e.g., in [52, 114]. Now, how can MPC be applied to \( (P) \)? At first, for chosen \( 0 \leq t_n < t^N_n \leq T \) with \( t^N_n = t_n + N \Delta T \) and \( y_n \in H \) we consider the following dynamical system on the time horizon \([t_n, t^N_n]\):

\[
y_n(t) = D(t; y(t), u(t)) \in V' \quad \text{a.e. in } (t_n, t^N_n], \quad y(t_n) = y_n \in H.
\] (3.1)

where the time-dependent mapping \( D(t; \cdot, \cdot) : V \times \mathbb{R}^m \to V' \) is defined as

\[
\langle D(t; \phi, u), \varphi \rangle_{V', V} = -a(t; \phi, \varphi) + \langle \mathcal{F}(t), \varphi \rangle_{V', V} + \sum_{i=1}^m u_i(t) \gamma_i \int_{\Gamma_c} b_i(s) \varphi(s) \, ds
\]

for \( \phi, \varphi \in V \), \( u = (u_i) \in \mathbb{R}^m \) a.e. in \([0, T]\). Furthermore, we consider the inequality constraints

\[
\begin{align*}
  u_{a_i}(t) \leq u_i(t) & \leq u_{b_i}(t), \quad i = 1, \ldots, m \quad \text{a.e. in } [t_n, t^N_n], \\
  y_a(t, x) \leq y(t, x) + \varepsilon w(t, x) & \leq y_b(t, x) \quad \text{a.e. in } Q_n = (t_n, t^N_n] \times \Omega, \\
  w_a & \leq w(t, x) \leq w_b \quad \text{a.e. in } Q_n.
\end{align*}
\] (3.2)

Next we define the function spaces related to \([t_n, t^N_n]\)

\[
U_n = L^2(t_n, t^N_n; \mathbb{R}^m), \quad W_n = L^2(Q_n), \quad X_n = W(t_n, t^N_n) \times U_n \times W_n.
\]

Moreover, let the set of admissible solutions be given as

\[
X_{ad}^{\varepsilon,n} = \{ x = (y, u, w) \in X_n \mid x \text{ satisfies (3.1) and (3.2)} \}.
\]

Now, the open-loop problem can be adapted by choosing the following cost

\[
J_n(x) = \frac{\sigma_Q}{2} \int_{t_n}^{t^N_n} \| y(t) - y_Q(t) \|^2_H \, dt + \frac{1}{2} \sum_{i=1}^m \sigma_i \| u_i \|^2_{L^2(t_n, t^N_n)} + \frac{\sigma_w}{2} \| w \|^2_{W_n}
\]

for \( x = (y, u, w) \in X_n. \)

**Remark 3.1.** Note that we choose \( \sigma_T = 0 \) for simplicity. This choice is due to the fact that there could be inconsistencies between the MPC optimal control subproblem and problem \( (P) \) if \( \sigma_T \neq 0 \). This can be avoided for example setting \( y_Q(t) \equiv y_T \) for \( t \geq T \). Because of the structure of the MPC Algorithm 9, this assumption will force the trajectory to remain close to \( y_T \) for every \( t \geq T \) and therefore in particular for \( t = T \). \( \diamond \)
Algorithm 9: MPC method

1: Choose initial state $y_0$, time horizon $N\Delta t$ and regularization parameter $\varepsilon > 0$;
2: Set $t_0 = 0$ and $y_0(0) = y_0$;
3: for $n = 0, 1, 2, \ldots$ do
4: Set $t_n^N = t_n + N\Delta t$;
5: Compute the optimal solution $\bar{x}_n^\varepsilon = (\bar{y}_n^\varepsilon, \bar{u}_n^\varepsilon, \bar{w}_n^\varepsilon)$ to
   $\min J_n(x)$ subject to $x \in X_{ad}^n; \quad \text{P}_n^\varepsilon$
6: Define the MPC feedback law $\Phi^N(t; y(t)) = \bar{u}_n^\varepsilon(t)$ for $t \in (t_n, t_n + \Delta t]$;
7: Get the MPC state $y_{n+1}$ solving
   $$(y)_t(t) = D(t; y(t), \Phi^N(t; y(t))) \in V'$ \ a.e. in $(t_n, t_n + \Delta t], \quad y(t_n) = y_n \in H;$$
8: Set $t_{n+1} = t_n + \Delta t$.
9: end for

If $\Phi^N$ is computed by the MPC algorithm, then the state $\bar{y}^N$ solves (3.1) for the closed-loop control $\bar{u}^N = \Phi^N(\cdot; y(\cdot))$ with a given initial condition $y_0$. Moreover, after a shift of the time horizon, Algorithm 9 always allows updating the data of the problem before Step 5. The next open-loop optimal control will be computed in such a way that the optimal closed-loop solution reacts to the changes of the parameters. On the contrary, this is not possible in the PDASS, since such online update will lead to divergence. It is clear that this possibility is a big advantage in a real application, such as room heating: suppose that we have a weather forecast for the next month telling us that the temperature will never be under $10^\circ C$, but unexpectedly the forecast changes after two weeks and the new predicted minimum temperature is $0^\circ C$. Clearly, the optimal control computed for the entire month by the PDASS will not be able to handle temperatures which are way lower than the expected one. On the other hand, due to its feedback structure, the MPC algorithm can react as soon as $y_{\text{out}}$ is updated according to the new weather forecast, implying that the MPC control will perform better than the PDASS one; cf. Section 3.3.1. Moreover, in this test case, the real optimal control could be computed only knowing in advance for the entire month the exact outside temperature. This is impossible with the weather forecast models available nowadays. Therefore, the MPC feedback control becomes relevant for our application. Furthermore, another asset for the MPC algorithm is
that the feedback control for the time interval \((t_n, t_n + \Delta t]\) is immediately accessible after each Step 6. Once again, there is no on the fly availability of the control for the PDASS. At last, the MPC subproblems are solved for a smaller number of time steps compared to the PDASS, helping with the curse of dimensionality in time, as pointed out in Remark 1.48. Nevertheless, also the MPC algorithm has its own cons. For example, it is well known that the larger the MPC horizon \(N\) is the better the MPC closed-loop control will approximate the optimal control in \([0, T]\). However, how large the horizon \(N\) has to be chosen to have a good approximation is not clear a-priori. In [51], for example, there are suboptimality estimates for the MPC cost functional \(J_n\) which gives the possibility to check a-posteriori how much the MPC suboptimal trajectory is close to the optimal one. These estimates can be used also for finding a suitable horizon \(N\); cf. [51]. Unfortunately, they can not be directly applied in our setting, since they are derived for autonomous dynamical systems and perfect tracking. The main problem is that in our case it is not clear if the system has always an equilibrium. Moreover, the cost to keep the trajectory at the equilibrium may not be zero. This implies that the relative performance estimates in [51] are going to zero as soon as the time increases. To avoid this issue, in [49] we introduce an absolute performance estimator, but still the estimate holds only for an autonomous system. In future works we are interested to extend the proofs to the example presented in this thesis. Due to this consideration, in Section 3.3, we assume that the horizon \(N\) is chosen sufficiently large to have a good approximation of the optimal control in \([0, T]\).

### 3.2 MPC-POD algorithms

In Chapter 2, the POD method is introduced to get rid of the high number of FE DOFs, constructing a reduced order model. This method can be used in Algorithm 9 to speed-up the computation of the solution to \((P_\epsilon^*)\) and, thus, to obtain a suboptimal control for the whole time horizon. As already mentioned, one has to take care of the quality of the POD approximation and the length of the horizon \(N\), in order to compute a control which is close enough to the optimal one, so that it can be accepted from an application point of view. As said, there is no a-priori and a-posteriori estimates available for \(N\) in our setting. Thus, we previously run tests for different \(N\) and we show the numerical results only for the smaller horizon such that its increase does not change significantly the new MPC optimal control. On the other hand, for the POD approximation we can check a-posteriori the quality
of the POD basis with the error estimates derived in Section 2.4. Moreover, as said, also the snapshots have a key role in this setting, since we would like to have a POD basis which can be used for as many MPC steps as possible. We want to point out that finding suitable snapshots is a really big challenge in this non-autonomous setting, since every open-loop optimal control starts with a different initial guess and the time-dependent parameters, in particular the advection field, may also introduce new dynamics in the problem. Luckily, the MPC algorithm itself is designed to have at the $n$-th step a knowledge of the future dynamic until the time $t_n^N$. Therefore, we can start solving the first open-loop problem with the full-order PDASS algorithm, if it is possible, and then use the open-loop trajectory as snapshot for the computation of the POD basis. We expect that for some time steps the quality of the approximation is good enough, since the first predicted open-loop trajectory would not be too far from the future dynamics, unless a parameter update is performed. Then, one can continue with the reduced order model, checking the a-posteriori error and refreshing the POD basis solving the full-order model when necessary. The previous strategy is resumed in Algorithm 10. Note that when we refresh the POD basis, we are also correcting the MPC-POD model, thanks to the full-order solve and the feedback structure of the MPC method.

**Algorithm 10** MPC-POD-PDASS method

1: Choose an initial state $y_0$, time horizon $N \Delta t$, regularization parameter $\varepsilon > 0$, tolerance $\tau_1$ for the a-priori estimate (2.11) and tolerance $\tau_2$ for the a-posteriori estimate;
2: Set $t_0 = 0$, $y_0(0) = y_0$, and $\text{flag} = \text{true}$;
3: for $n = 0, 1, 2, \ldots$ do
4: Set $t_n^N = t_n + N \Delta t$;
5: if $\text{flag} = \text{true}$ then
6: Compute the optimal solution $\bar{x}_n^\varepsilon = (\bar{y}_n^\varepsilon, \bar{u}_n^\varepsilon, \bar{w}_n^\varepsilon)$ to $\mathcal{P}_n^\varepsilon$;
7: Compute a POD basis $\{\psi_i\}_{i=1}^\ell$ of rank $\ell$ according to $\tau_1$ using as snapshots $\bar{y}_n^\varepsilon$ and $\bar{p}_n^\varepsilon$;
8: Define the MPC feedback law $\Phi(t; y(t)) = \bar{u}_n^\varepsilon(t)$ for $t \in (t_n, t_n + \Delta t]$;
9: Set $\text{flag} = \text{false}$;
else
10: Compute the POD suboptimal solution $\bar{x}_n^{\varepsilon,\ell} = (\bar{y}_n^{\varepsilon,\ell}, \bar{u}_n^{\varepsilon,\ell}, \bar{w}_n^{\varepsilon,\ell})$ to $\mathcal{P}_n^\varepsilon$;
11: Define the MPC feedback law $\Phi(t; y(t)) = \bar{u}_n^{\varepsilon,\ell}(t)$ for $t \in (t_n, t_n + \Delta t]$;
12: Compute the a-posteriori error estimate $e := \|T^*\zeta\|_2$ from (2.14);
if \( e > \tau_2 \) then
Set \( \text{flag} = \text{true} \);
end if
Get the MPC state \( y_{n+1} \) solving
\[
(y)(t) = D(t; y(t), \Phi^N(t; y(t))) \in V' \text{ a.e. in } (t_n, t_n + \Delta t], \quad y(t_n) = y_n \text{ in } H;
\]
Set \( t_{n+1} = t_n + \Delta t \).
end for

A possible drawback in Algorithm 10 is that we have to perform several full-order solves, if we have to update the POD basis frequently. This procedure increases significantly the computational time. Moreover, due to the curse of space dimensionality, it may be not even possible to use the full-order PDASS algorithm, unless we exploit its matrix-free structure. Therefore, we propose a second less efficient way to generate the snapshots. For example, we can use the augmented Lagrangian method, which requires only matrices whose size is the number of finite element nodes. Now, there are several possibilities to make use of the ALM. In Chapter 1 we proposed to set a tolerance \( \epsilon \) for the residuals to stop the algorithm. Of course, the smaller is \( \epsilon \) the better the solutions will be, but more time is needed to compute it. This technique has also a second advantage: make use of the augmented Lagrangian at first will produce a good warm start for the semi-smooth Newton method, decreasing the number of iterations to converge for smaller regularization parameter \( \epsilon \). In fact, as shown in Section 1.5, the local convergence of the semi-smooth Newton is affected by \( \epsilon \), the smaller \( \epsilon \) is the closer we are to the solution of (1.15). Hence, the admissible set of controls is a subset of the one belonging to the problem with larger value of \( \epsilon \). Due to this consideration, one has to be careful in the choice of the stopping criterion for the ALM method, i.e., this has to be done according to \( \epsilon \). In fact, if we stop the algorithm too early, the ALM sub-optimal solution may not be a good warm start. On the other hand, if we stop too late, the ALM sub-optimal solution may be inside the bounds with almost all points, but the controls related to this solution may be too high compared to the ones which are solution of the problem regularized with the VC. In our numerical simulations, we set the tolerance \( \epsilon \) such that the residuals \( R_a^{n-1} \) and \( R_b^{n-1} \) are not so large compared to the desired violation of the state constraints for the VC. All the previous considerations are resumed in Algorithm 11, where the MPC-POD-ALM-PDASS algorithm is described.
Algorithm 11 MPC-POD-ALM-PDASS method

1: Choose an initial state \( y_0 \), time horizon \( N \Delta t \), regularization parameter \( \varepsilon > 0 \), tolerance \( \tau_1 \) for the a-priori estimate (2.11) and tolerance \( \tau_2 \) for the a-posteriori estimate;
2: Set \( t_0 = 0 \), \( y_0(0) = y_0 \), and \( \text{flag} = \text{true} \);
3: Set ALM tolerance \( \tau_3 \) for the stopping criterion, initial penalty parameter \( c_0 \) and increment \( \beta \) according to \( \varepsilon \);
4: for \( n = 0, 1, 2, \ldots \) do
5: \( t^N_n = t_n + N \Delta t \);
6: if \( \text{flag} = \text{true} \) then
7: Run Algorithm 3 setting \( \varepsilon = \tau_3 \);
8: Compute a POD basis \( \{ \psi_i \}_{i=1}^\ell \) of rank \( \ell \) according to \( \tau_1 \) using as snapshots the scaled \( y_n \) and \( p_n \) of the ALM method;
9: Compute with PDASS the POD suboptimal solution \( \bar{x}_n^{\varepsilon,\ell} = (\bar{y}_n^{\varepsilon,\ell}, \bar{u}_n^{\varepsilon,\ell}, \bar{w}_n^{\varepsilon,\ell}) \) to \( (P'_n) \) using as warm start the suboptimal control computed in Step 7;
10: Define the MPC feedback law \( \Phi_N^{\varepsilon}(t; y(t)) = \bar{u}_n^{\varepsilon,\ell}(t) \) for \( t \in (t_n, t_n + \Delta t] \);
11: Set \( \text{flag} = \text{false} \);
12: else
13: Compute with PDASS the POD suboptimal solution \( \bar{x}_n^{\varepsilon,\ell} = (\bar{y}_n^{\varepsilon,\ell}, \bar{u}_n^{\varepsilon,\ell}, \bar{w}_n^{\varepsilon,\ell}) \) to \( (P'_n) \);
14: Define the MPC feedback law \( \Phi_N^{\varepsilon}(t; y(t)) = \bar{u}_n^{\varepsilon,\ell}(t) \) for \( t \in (t_n, t_n + \Delta t] \);
15: Compute the a-posteriori error estimate \( e := \| T^* \zeta \|_Z \) from (2.14);
16: if \( e > \tau_2 \) then
17: Set \( \text{flag} = \text{true} \);
18: end if
19: end if
20: Get the MPC state \( y_{n+1} \) solving
\[
(y)_t(t) = D(t; y(t), \Phi_N(t; y(t))) \in V' \text{ a.e. in } (t_n, t_n + \Delta t], \quad y(t_n) = y_n \text{ in } H;
\]
21: Set \( t_{n+1} = t_n + \Delta t \).
22: end for

Remark 3.2 (Snapshots selection). Everytime the POD basis is updated, we compute new snapshots, since we need to add information about the new dynamics involved. What about the old information? One can think to make use of
the previous snapshots to compute the new POD basis, since this information can be helpful to improve the quality of the approximation. A-priori this is a good idea, but it may happen that we are including information, which are not needed to compute an accurate reduced-order solution for the next MPC steps. Moreover, in this setting, it may become inconvenient also for another reason: everytime we compute a new basis, if we want to keep the a-priori estimate (2.6) below a certain tolerance, we have to choose a higher rank \( \ell \) compared to the previous one, implying that the dimension of the reduced order model increases; cf. Section 3.3. An increasing \( \ell \) for a large (or infinite) \( T \) may overtake the number of DOFs \( N_x \), making useless the reduced order model. Therefore, in Algorithm 12 we propose a snapshots selection procedure based on the consideration that the discrete trajectories can be seen as vectors of \( \mathbb{R}^{N_x} \) and that the new computed snapshots are the starting point to build the new POD basis. Due to Step 4 in Algorithm 12, we discard essentially the dynamics which are "not close enough" or "too similar" to the new computed ones, i.e., the ones which do not help to decrease the approximation error of the reduced-order model. As shown in Section 3.3, this procedure helps to improve the reduced order model keeping \( \ell \) small with properly chosen tolerances \( \rho, \varrho \) and inner product \( \langle \cdot, \cdot \rangle \).

**Algorithm 12** Snapshots selection

**Require:** Snapshots previously computed and stored in a list \( L \) and tolerances \( 0 < \rho < \varrho << 1 \).

1. Compute the new \( M \) snapshots and store them in a list \( S \);
2. for \( i \leq \text{length}(L) \) do
3. for \( j \leq M \) do
4. if \( (1 - \varrho) \| S[j] \| \| L[i] \| \leq |\langle S[j], L[i] \rangle| \leq (1 - \rho) \| S[j] \| \| L[i] \| \) then
5. Add \( L[i] \) to \( S \);
6. break
7. end if
8. end for
9. end for

### 3.3 Numerical tests

In this section, we study the efficiency of the proposed MPC-POD algorithms. At first, in Section 3.3.1 we compare the MPC algorithm to the PDASS, showing the advantages connected to computing a feedback control. Subsequently, we test
3.3 Numerical tests

the proposed combination of MPC, POD, PDASS and a-posteriori error estimator, verifying the computational time speed-up and the suboptimality of the reduced-order solution. Moreover, we show how considering a basis update strategy together with Algorithm 12 improves the results.

3.3.1 Feedback versus open-loop control

As already mentioned, the PDASS alone can not take care of the changes of the parameters. Therefore, it is necessary to combine it with a MPC strategy. In this section, we show with a numerical example how much a combination of MPC and PDASS can improve the results of the PDASS used alone. Regarding the state equation, we choose the same setting as in Section 1.5 except for $\gamma_o = 0.2$. Hence, the effect of the outside temperature on the boundary $\Gamma_o$ is stronger than the one in Section 1.5. Furthermore, we fix as MPC prediction horizon $N = 60$ and we run our simulation in the time interval $[0, 4]$. Aware of the results reported in Table 1.3, we fix $\varepsilon = 0.01$, since for our application small violations of the constraints are accepted. In fact, small temperature differences in few points of the room can not be detected by the human body, allowing such relaxation of the state constraints. Moreover, for the optimal control problem, we choose a different upper bound for the state, i.e.

$$y_b(t, x) = \begin{cases} 
25 & t \in [0, 1], \ x \in \overline{\Omega}, \\
26 - t & t \in (1, 3], \ x \in \overline{\Omega}, \\
23 & t > 3, \ x \in \overline{\Omega}, 
\end{cases}$$

(3.3)

fixing the rest of the parameters as the one in Section 1.5.2. Since $\sigma_T = \sigma_Q = 0$, we are in the setting of economic MPC. The choice of a different $y_b$ respect to the previous chapters is motivated by the application: at the begin we empower the control to keep the temperature almost inside the bounds, then, to avoid uncomfortable high temperature in the room, we decrease progressively the upper bound. At last, we solve the PDASS algorithm with $y_{out}(t) = 16.0$ for all $t \in [0, 4]$. For the MPC-PDASS algorithm we suppose instead that at the time instance $t = 2$ we have a new weather forecast, so that from $t = 2$ we have $y_{out}(t) = \max(16.0 - t, 10.0)$. We expect that the MPC-PDASS algorithm will react to this change. Instead, since in the PDASS algorithm there is no possibility to update parameters during the optimization procedure, we predict that the computed optimal control will not be good for this new scenario. In Figure 3.1, we report the solution to the dynamical system (3.1) computed using the PDASS control and the
MPC-PDASS one for

\[ y_{out}(t) = \begin{cases} 
16 & t \in [0, 2], \\
16 - t & t \in (2, 4]. 
\end{cases} \quad (3.4) \]

Of course, both solutions are suboptimal, since the PDASS control is optimal only when \( y_{out} = 16 \) for all the time steps and the MPC-PDASS one does not know that the outside temperature changes until the time step \( t = 2 \). Anyway, the MPC-PDASS algorithm reacts and tries to keep the temperature inside the bounds, as it can also be seen from Figure 3.2, where the controls are shown. Clearly, if the parameter is updated too late and it differs much from the previous one, the MPC algorithm may take time before the solution is brought again inside the range of the state constraints. Thus, for some time steps our solution may not be feasible for problem (1.15). For example, suppose that you are driving a car with constant speed and your braking distance is ten meters. If five meters in front of you there is a traffic light that immediately turns red, obviously you can not stop in time. Similarly, the MPC reaction can not avoid this infeasibility of the solution. However, since in our numerical example we have chosen \( \varepsilon = 0.01 \) allowing small violations of the constraints, the MPC-PDASS algorithm can still compute a suboptimal solution, which is better than the PDASS one computed not knowing that \( y_{out} \) has changed.

As last, to validate our thesis, we point out that the MPC-PDASS cost functional is equal to 4457.91 and the PDASS one is 35477.98. These values are computed considering the norm of the controls and of the violation of the state constraints.
Figure 3.2: MPC-PDASS control reaction to the update of the parameter $y_{\text{out}}$.

Remark 3.3. We observe a trade-off among the violations of lower and upper bounds in Figure 3.1(b). As our example is designed, the reaction of the controls is also mitigated by the constraint $y_b$. In fact, this reaction takes in account that the maximum temperature in the room can not exceed too much the upper bound $y_b$. Thus, the possibility to obtain infeasible solutions rises. This suggests different approaches to fix this problem. For example, one could think to impose state constraints only in some regions of the domain $\Omega$, excluding the boundary parts where the control is acting. Another possibility is to consider an integral constraint as in [65], then the goal is to control the average temperature, making also the optimal control problem easier to be solved. These considerations can be a starting point for future works.

3.3.2 MPC and POD

For these numerical tests we refer to the test data in Section 1.5. Moreover, we set the prediction horizon $N = 60$ and we stop the MPC algorithm after $T = 4.0$. For the same reasons of Section 3.3.1, we set $\varepsilon = 0.01$ and $y_b$ as in (3.3), keeping the other as in Section 1.5.2. Therefore, we are again in the setting of economic MPC. In the first part of this section we compare the MPC-FE solution with the reduced-order one (MPC-POD-EE) computed by Algorithm 10. Moreover, we report the results obtained with the MPC-POD algorithm without updating the POD basis (MPC-POD), i.e., setting a too large $\tau_2$ in Algorithm 10. Every time we compute a POD basis, we select $\ell$ according to (2.11) with tolerance $\tau_1 = 10^{-8} \sum_{i=1}^{D} \lambda_i$. Already
from Figure 3.3, one can deduce that the solution computed updating the POD basis
during the MPC algorithm is closer to the full-order one than the one computed
using the same reduced-order model for all the steps of Algorithm 10. This confirms
what shown in Chapter 2: it is extremely important to certify the approximation
error of the reduced-order model. Moreover, according again to Figure 3.3, the
room temperature can be kept inside the bounds with small violations, which are
irrelevant for our application. In particular, the average temperature in the room is
at every time step inside the constraints range. Therefore, the proposed approach,
i.e., to fix a small enough $\varepsilon$ instead of approximating the solution to (1.15), can
be considered valid. The same remarks can be done looking at Table 3.1. In fact,

![Graphs showing optimal solution maximum, minimum and average values and constraints](image)

(a) MPC-FE  
(b) MPC-POD  
(c) MPC-POD-EE

**Figure 3.3:** Plot of optimal solution maximum, minimum and average values and
of the constraints $y_a$ and $y_b$. 
3.3 Numerical tests

<table>
<thead>
<tr>
<th>Method</th>
<th>$J$</th>
<th>rel_err_u</th>
<th>rel_err_y</th>
<th>State Act. Pts.</th>
<th>rel_err_AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-FE</td>
<td>2593.58</td>
<td>–</td>
<td>–</td>
<td>4054 (2.2%)</td>
<td>–</td>
</tr>
<tr>
<td>MPC-POD</td>
<td>2597.21</td>
<td>0.039</td>
<td>0.0145</td>
<td>3039 (1.7%)</td>
<td>0.322</td>
</tr>
<tr>
<td>MPC-POD-EE</td>
<td>2593.75</td>
<td>0.002</td>
<td>0.0003</td>
<td>4090 (2.3%)</td>
<td>0.016</td>
</tr>
</tbody>
</table>

Table 3.1: Results for Algorithm 9 (MPC-FE) and Algorithm 10 with fixed POD basis (MPC-POD) and with basis updates (MPC-POD-EE).

the number of state active points is only around the 2% for all the methodologies. Furthermore, the reported relative errors for the optimal control (rel_err_u) and for the optimal state (rel_err_y) show that using the error estimator to update the POD basis improves the results. This can be also deduced from Figure 3.4, where the absolute difference between MPC-FE and MPC-POD controls with and without the basis updates is shown. Also the values of the cost functionals shown in Table 3.1 underline the improvement given by the a-posteriori error estimator: the value of $J$ for MPC-POD-EE is one order of magnitude closer than the MPC-POD one. Looking again to the number of state active points, it seems that MPC-POD unexpectedly performs better than the others, since less points are active. Actually, as one can see from the values of the cost functionals reported in Table 3.1 and from Figure 3.3, the MPC-POD solution violates the constraints more than MPC-FE and MPC-POD-EE ones. Therefore, MPC-FE optimal solution performs better than the other two methodologies, as predicted by the theory. The importance to

![Figure 3.4](image-url)

(a) MPC-POD  (b) MPC-POD-EE

Figure 3.4: Control difference $\delta_i := |\bar{u}_i^\varepsilon - \bar{u}_i^{\varepsilon,\ell}|$ at each time step for $i = 1, \ldots, m$. 
update the basis for improving the results is also confirmed by rel_err_AS, reported in Table 3.1 and defined in Section 2.5. In fact, for this test, the MPC-POD relative error for the state active sets is twenty times larger than the MPC-POD-EE one. Of course, there are not exclusively advantages in using the a-posteriori error estimator. As one can see from Table 3.2, the computational speed-up of MPC-POD-EE is significantly reduced compared to the one of MPC-POD. This is due to the fact that we have to compute the full-order open-loop optimal control for each POD basis update, according to Algorithm 10. Moreover, evaluating the error estimator implies solving full-order state and adjoint equations; cf. Section 2.4.1. At last, at every basis update the number of POD basis functions increases as shown in Table 3.3. The reason relies on the fact that both old and new snapshots are used to compute a new POD basis, i.e., the snapshots subspace is enriched with new dynamics. Therefore, the number of basis may increase to keep the same a-priori tolerance \( \tau_1 \). For example, at the last refresh we have a snapshots set of dimension \( D = 600 \) with a large variety of dynamics, thus the algorithm select \( \ell = 186 \) basis functions, because of the small chosen tolerance \( \tau_1 \). Due to this last consideration,

\[
\begin{array}{c|c|c|c|c}
\text{Basis updates} & 0 & 1 & 2 & 3 & 4 \\
\text{Basis functions} & 65 & 103 & 132 & 165 & 186 \\
\end{array}
\]

Table 3.3: Number of POD basis functions after each POD basis update according to tolerance \( \tau_2 \) and a-posteriori error estimator (2.14).

we apply the snapshots selection procedure explained in Algorithm 12. We do a first test (MPC-POD-EED) choosing \( \rho = 0.001 \) and \( \varrho = 0.005 \) and a second one (MPC-POD-EED2) with \( \rho = 0.001 \) and \( \varrho = 0.2 \). Moreover, we perform a test where we use only the new computed snapshots for each basis update, throwing away the old ones (MPC-POD-EEONS). In this last test, we expect that the number of basis functions will be around the same order of magnitude for every update, since

<table>
<thead>
<tr>
<th>Method</th>
<th>Sol. avg. time</th>
<th>Sol. speed-up</th>
<th>Tot. time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-FE</td>
<td>26.73 s</td>
<td>-</td>
<td>3.54 h</td>
<td>-</td>
</tr>
<tr>
<td>MPC-POD</td>
<td>0.49 s</td>
<td>54.55</td>
<td>1.06 h</td>
<td>3.34</td>
</tr>
<tr>
<td>MPC-POD-EE</td>
<td>2.12 s</td>
<td>12.61</td>
<td>2.40 h</td>
<td>1.48</td>
</tr>
</tbody>
</table>

Table 3.2: Average computational time to perform a PDASS iteration, total computational time for the MPC algorithms and POD speed-up.
the dimension of the snapshots subspace remains constant. On the other hand, excluding the old snapshots may not guarantee a POD approximation as good as MPC-POD-EE. At first, Table 3.4 shows that the snapshots selection does not affect the approximation error and the number of active points. As expected, discarding the dynamics which are far from the optimal ones does not change the behaviour of Algorithm 10. Furthermore, it improves the computational time according to Table 3.5. The reason is shown in Figure 3.5: thanks to the selection strategy the number of POD basis functions does not always increase, but it may happen that we need more basis updates. This fact could slow down the computation and affect the improvement in computational time. This occurs for MPC-POD-EED, where also important dynamics are discarded, due to the strict set tolerance \( \varrho \). When we increase \( \varrho \) (MPC-POD-EED2), we are including more snapshots, having a greater number of basis, but without missing important information. Thus, less POD basis updates are needed and this improves the computational time. Due to the previous considerations, it makes sense to examine in depth and improve the snapshots selection strategy in future works. At last, MPC-POD-EEONS results show that it is important to include the old snapshots (at least for this test). Even

<table>
<thead>
<tr>
<th>Method</th>
<th>( J )</th>
<th>( \text{rel.err}_u )</th>
<th>( \text{rel.err}_y )</th>
<th>State Act. Pts.</th>
<th>( \text{rel.err}_\text{AS} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-POD-EED</td>
<td>2594.07</td>
<td>0.003</td>
<td>0.0002</td>
<td>4095 (2.3%)</td>
<td>0.013</td>
</tr>
<tr>
<td>MPC-POD-EED2</td>
<td>2593.59</td>
<td>0.003</td>
<td>0.0004</td>
<td>4100 (2.3%)</td>
<td>0.016</td>
</tr>
<tr>
<td>MPC-POD-EEONS</td>
<td>2594.14</td>
<td>0.003</td>
<td>0.0003</td>
<td>4095 (2.3%)</td>
<td>0.013</td>
</tr>
</tbody>
</table>

Table 3.4: Results for Algorithm 10 either excluding the old snapshots (MPC-POD-EEONS) or performing the snapshots selection with tolerances \( \rho = 0.001 \) and \( \varrho = 0.05 \) (MPC-POD-EED) or \( \varrho = 0.2 \) (MPC-POD-EED2).

<table>
<thead>
<tr>
<th>Method</th>
<th>Sol. avg. time</th>
<th>Sol. speed-up</th>
<th>Tot. time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-FE</td>
<td>26.73 s</td>
<td>–</td>
<td>3.54 h</td>
<td>–</td>
</tr>
<tr>
<td>MPC-POD-EED</td>
<td>1.71 s</td>
<td>15.63</td>
<td>1.88 h</td>
<td>1.88</td>
</tr>
<tr>
<td>MPC-POD-EED2</td>
<td>1.20 s</td>
<td>22.28</td>
<td>1.47 h</td>
<td>2.41</td>
</tr>
<tr>
<td>MPC-POD-EEONS</td>
<td>23.67 s</td>
<td>1.13</td>
<td>3.32 h</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Table 3.5: Average computational time to perform a PDASS iteration, total computational time for the MPC algorithms and POD speed-up with snapshots selection or excluding the old snapshots.
if in Table 3.4 the relative errors are of the same order of the other methods, there is almost no time speed-up (cf. Table 3.5). Figure 3.5 shows the reason: every two or

![Figure 3.5: Number of POD basis functions at each time step for MPC-POD-EE, MPC-POD-EED, MPC-POD-EED2 and MPC-POD-EEONS. (■ = POD basis update.)](image)

three time steps the POD basis are updated, because only the new information are not enough to guarantee a good error approximation, according to the a-posteriori estimator. Therefore, for half of the time steps we are solving the optimal control problem with FE, obtaining a good approximation of the optimal control $\bar{u}$, but with almost the same computational time of MPC-FE. As last numerical simulation, we test Algorithm 11, keeping the same parameters. We set $\tau_3 = 0.5$ as stopping criterion for the ALM algorithm. This tolerance is not too far from the expected and desired violation of the state constraints. Furthermore, aware of the previous results, we apply the snapshots selection strategy with tolerances $\rho = 0.001$ and $\varrho = 0.2$ (MPC-POD-EED2 and MPC-POD-ALM respectively).

<table>
<thead>
<tr>
<th>Method</th>
<th>$J$</th>
<th>rel_err$_u$</th>
<th>rel_err$_y$</th>
<th>State Act. Pts.</th>
<th>rel_err AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-POD-EED2</td>
<td>2593.59</td>
<td>0.003</td>
<td>0.0004</td>
<td>4100 (2.3%)</td>
<td>0.016</td>
</tr>
<tr>
<td>MPC-POD-ALM</td>
<td>2594.16</td>
<td>0.004</td>
<td>0.0005</td>
<td>4105 (2.3%)</td>
<td>0.013</td>
</tr>
</tbody>
</table>

Table 3.6: Results for Algorithm 10 and for Algorithm 11 performing the snapshots selection with tolerances $\rho = 0.001$ and $\varrho = 0.2$ (MPC-POD-EED2 and MPC-POD-ALM respectively).
can see from Table 3.6, the results are similar to the one obtained with Algorithm 10. Anyway, we want to point out that the number of basis updates is doubled respect to the one of Algorithm 10. We have, in fact, 9 POD basis updates. This is expected, because the snapshots computed by the ALM are suboptimal compared to the one computed by the PDASS algorithm. Therefore, especially at the begin, the quality of the POD approximation is worse than the previous simulations. This is confirmed by Table 3.7, where we report at which time step a POD basis update is performed and how many POD basis functions are selected. As one can see, this updates are mainly performed at the first time instances, meaning that the suboptimal ALM snapshots are not good enough to guarantee a small approximation error at the begin. This

<table>
<thead>
<tr>
<th>Time</th>
<th>0.0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.7</th>
<th>0.8</th>
<th>1.0</th>
<th>1.8</th>
<th>2.2</th>
<th>3.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basis functions</td>
<td>65</td>
<td>113</td>
<td>75</td>
<td>124</td>
<td>160</td>
<td>174</td>
<td>123</td>
<td>117</td>
<td>106</td>
</tr>
</tbody>
</table>

Table 3.7: POD basis functions for each basis update and time at which this update occurs for MPC-POD-PDASS-ALM algorithm

influences the computational time: it turns out that the MPC-POD-ALM-PDASS algorithm is not faster compared to the MPC-POD-PDASS one, as one can notice in Table 3.8. One can think then to decrease the tolerance $\tau_3$ for the residuals $R_{\text{a}}^{n,+}$ and $R_{\text{b}}^{n,+}$ to improve the snapshots and decrease the number of updates, but this will cause, of course, an increase of the computational time, since we need to perform more ALM iterations with the full-order model. Therefore, it is really important to find a good compromise between the ALM and the VC. If the number of snapshots updates is significantly reduced using only few more iterations of the full-order ALM algorithm, then there is the possibility to speed-up the procedure. Unfortunately, the challenge to find a good interplay between the proposed techniques is not easily solvable. A more rigorous way has to be found to choose the tolerance $\tau_3$ for the

<table>
<thead>
<tr>
<th>Method</th>
<th>Sol. avg. time</th>
<th>Sol. speed-up</th>
<th>Tot. time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-FE</td>
<td>26.73 s</td>
<td>–</td>
<td>3.54 h</td>
<td>–</td>
</tr>
<tr>
<td>MPC-POD-EED2</td>
<td>1.20 s</td>
<td>22.28</td>
<td>1.47 h</td>
<td>2.41</td>
</tr>
<tr>
<td>MPC-POD-ALM</td>
<td>3.36 s</td>
<td>7.96</td>
<td>1.49 h</td>
<td>2.38</td>
</tr>
</tbody>
</table>

Table 3.8: Average computational time to perform a PDASS iteration, total computational time for the MPC algorithms and POD speed-up with snapshots selection and ALM suboptimal snapshots.
computation of the snapshots. We believe that there is the possibility to establish a mathematical relation between the regularization parameter $\varepsilon$ for the VC and the residuals $R^n_{a,+}$ and $R^n_{b,+}$ of the ALM. Such relation will help in the a-priori choice of the stopping criterion of the ALM snapshots generation, improving the results of Algorithm 11. This will be investigated in future works.
Conclusion

In this thesis we presented regularization and penalization techniques for pointwise state-constrained linear-quadratic optimal control problem: the virtual control approach and the augmented Lagrangian method, respectively. We applied them to the particular case of a parabolic convection-diffusion equation. Moreover, existing proofs for different problems [79, 80, 83] were adapted to our specific setting, showing convergence for the previously mentioned techniques. The main differences to [79, 80, 83] rely on bilateral pointwise state constraints, parabolic convection-diffusion equation and Robin boundary controls. A model order reduction approach based on POD was applied to speed-up the computational time. Furthermore, an a-posteriori estimate for the approximation error of the reduced order model was derived for the virtual control approach and the existing estimate [127] was considered for the augmented Lagrangian method. These estimates were used to certify the POD approximation error and to develop a POD basis update procedure. Moreover, an economic MPC strategy was employed to take in account the changes of problem parameters in a feedback way. The proposed methodologies were successfully combined with a particular focus on an application to energy efficient building.

Regarding future work, although we have already addressed throughout the thesis specific tasks which can be object of study, we make general remarks in what follows. At first, the model can be improved. In the thesis we considered a linear convection-diffusion equation, where the velocity field $v$ solves the incompressible Navier-Stokes equation. Therefore, a possible improvement is to consider the full coupling of the convection-diffusion equation with the Navier-Stokes in the so-called Boussinesq approximation [66, 113]. Despite the increase of theoretical and numerical challenges, the Boussinesq effect, i.e., the physical phenomenon for which the airflow is influenced by temperature differences, is perfectly designed for central heating. As an intermediate step, one can consider the following inexact
model: the POD snapshots and the convection field can be generated solving the Boussinesq approximation for an arbitrary control, then the MPC-POD-PDASS algorithm for the linear-quadratic optimal control problem can be used to compute the optimal control, checking the POD approximation error with the a-posteriori error estimate. If an update of the POD basis is necessary, then we compute new snapshots and the velocity field through the Boussinesq model, using the stored closed-loop and the last computed open-loop controls. Thanks to the feedback property of the MPC algorithm, we expect that the gap between the computed suboptimal closed-loop controls and the optimal ones is reduced as soon as the time passes. Another important issue is the computation of the POD snapshots through the PDASS algorithm. To fix this problem we have already shown in this thesis the possibility to consider ALM suboptimal solutions, in order to get a good compromise between speed-up and accuracy, even if the interplay between the two techniques is difficult to be balanced. Another perspective is to consider domain decomposition preconditioners for the linear system involved at each PDASS iteration. For example, one can consider [18, 60] as starting point, although the authors do not consider state constraints. At last, to deal with the costly evaluation of the state active points, one can consider localized model order reduction techniques [106, 107], in order to split the domain in several subdomains and building different reduced-order models with a different number of basis functions or even using a FE discretization in some of them. This procedure should speed up the computation of the solution, giving good performances in terms of approximation. The previous ideas are some of the possible developments of this work and they will be the main focus of our future research.
Bibliography


