



---

*Research article*

## Douglas–Rachford algorithm for control- and state-constrained optimal control problems

Regina S. Burachik\*, Bethany I. Caldwell and C. Yalçın Kaya

Mathematics, UniSA STEM, University of South Australia, Mawson Lakes, S.A. 5095, Australia

\* **Correspondence:** Email: [regina.burachik@unisa.edu.au](mailto:regina.burachik@unisa.edu.au).

**Abstract:** We consider the application of the Douglas–Rachford (DR) algorithm to solve linear-quadratic (LQ) control problems with box constraints on the state and control variables. We have split the constraints of the optimal control problem into two sets: one involving the ordinary differential equation with boundary conditions, which is affine, and the other, a box. We have rewritten the LQ control problems as the minimization of the sum of two convex functions. We have found the proximal mappings of these functions, which we then employ for the projections in the DR iterations. We propose a numerical algorithm for computing the projection onto the affine set. We present a conjecture for finding the costates and the state constraint multipliers of the optimal control problem, which can, in turn, be used to verify the optimality conditions. We conducted numerical experiments with two constrained optimal control problems to illustrate the performance and the efficiency of the DR algorithm in comparison with the traditional approach of direct discretization.

**Keywords:** optimal control; Douglas–Rachford algorithm; harmonic oscillator; control constraints; state constraints; numerical methods

**Mathematics Subject Classification:** 34H05, 49M37, 49N10, 65K10, 65L10

---

### 1. Introduction

Many of the problems we encounter in the world that can be presented as optimal control problems contain constraints on both the state and control variables. Imagine a scheduling problem whereby the aim is to maximize profits, such as in [26], in which the control variable is the production rate and the state variable is the inventory level. Naively, one may want to greatly increase both the inventory level and production rate to maximize profits, but, in practice, there is a limit to the amount of inventory a factory can hold, as well as a limit on how quickly humans or machines can work. In order to accurately model this problem, and many others from a wide range of application areas, such as manufacturing, engineering, science, economics, etc., we must use an optimal control problem with both state and

control constraints. Although these problems are commonly found in applications, they are much more difficult to solve than those optimal control problems which purely have control constraints.

We focus our attention on linear-quadratic (LQ) state- and control-constrained optimal control problems. These are infinite-dimensional optimization problems that involve the minimization of a quadratic objective function that is subject to linear differential equation (DE) constraints, affine constraints on the state variables, and affine constraints on the control variables. There is extensive literature on LQ control problems, as they can model many problems from a wide variety of disciplines (see [1, 14–16, 24, 27, 28]). Though many of the LQ control problems posed in the literature contain control constraints along with the linear DE constraints, it is rarer to see state constraints included since, as mentioned above, they are much more difficult to manage. Here, we propose a unique approach for solving state-constrained problems by applying the Douglas–Rachford (DR) algorithm.

The DR algorithm is a projection algorithm that is used to minimize the sum of two convex functions. In order to apply the algorithm one needs the proximal operators of the two convex functions. Splitting and projection methods such as the DR algorithm constitute a popular area of research in optimization, with a variety of applications (see [2, 3, 6, 9, 20]) for their use in sphere packing problems, protein reconstruction, etc. The use of these methods to solve discrete-time optimal control problems is not new, but there are very few applications of these methods to continuous-time optimal control problems. In [7], projection methods were used to solve the energy-minimizing double integrator problem with control constraints. Other papers [11–13] address more general energy-minimizing LQ control problems with control constraints. Additionally, a collection of general projection methods is used in [11], and the DR algorithm is used in [12, 13].

Following the promising numerical results observed in [12, 13], here, we have used the DR algorithm to address the more challenging control- and state-constrained LQ control problems.

The current technique for solving these control- and state-constrained LQ problems is a direct discretization approach whereby the infinite-dimensional problem is reduced to a large-scale finite-dimensional problem by using a discretization scheme, e.g., a Runge–Kutta method [5, 22]. This discretized problem is then solved through the use of an optimization modelling language such as AMPL [19], paired with (finite-dimensional) large-scale optimization software such as Ipopt [33].

Through the present paper (as was done in [7, 11] for control constraints only), we present the following contributions to research on the numerical solution of LQ control problems with state and control constraints.

- (i) We rewrite the original LQ control problem as the minimization of the sum of two convex functions. This reformulation addresses much more general problems than those considered in [7, 11]. Our setting encompasses important practical problems, such as those referenced above (i.e., problems that have constraints on the state variables).
- (ii) We find the proximal mappings of these functions by solving the resulting calculus of variations subproblems (Theorems 1 and 2). Although the proof of Theorem 1 is straightforward thanks to the separability of the variables  $x$  and  $u$ , the proof of Theorem 2 is more involved due to a new LQ control subproblem that needs to be solved.
- (iii) We define and apply the DR algorithm to the constrained LQ control problem. We define and calculate proximal mappings at  $(x, u)$ , rather than just  $u$ , as in [7, 12]. Thus, we describe the DR algorithm for the iterate  $(x^k, u^k)$  rather than for  $u^k$  alone.
- (iv) We propose an algorithm for computing the projection of an iterate  $(x^-, u^-)$  onto the affine

constraint set defined by the ordinary differential equations (ODEs). The procedure is an extension of [12, Algorithm 2], which projects only  $u^-$ . The current algorithm involves a more complicated two-point boundary-value problem to solve.

- (v) We perform numerical experiments to illustrate the higher efficiency of the DR algorithm as compared to the traditional approach of direct discretization. Solving optimal control problems with state constraints is well known to be more challenging than those without them. We demonstrate the value of the new approach in solving a class of such problems.

In addition to these, we also present the following contribution, which is new even for the purely control-constrained LQ control problems.

- (vi) We present a conjecture for finding the costates. Moreover, we present a technique for finding the multiplier for the state constraints in the case that only one state constraint is active at a given time. We have effectively employed the costates and the state control multiplier for numerical verification of the optimality conditions.

The two convex functions mentioned above are defined as follows: the information from the ODE constraints appears in one function, while the information from the control and state constraints, along with the objective functional appear in the other. We derive the proximal mappings of the two convex functions without reducing the original infinite-dimensional optimization problem to a finite-dimensional one, though we need to discretize the state and control variables over a partition of time when implementing the DR algorithm since a digital computer cannot iterate with functions.

The paper is structured as follows. In Section 2, we give the problem formulation and optimality conditions for the optimal control problem. In Section 3, we derive the proximal mappings used in the implementation of the DR algorithm. Section 4 introduces the DR algorithm. Then, Section 5 begins with an algorithm for computing one of the proximal mappings (namely, the projection onto an affine set) and a conjecture that is used to obtain the costate variable of the original LQ control problem. Next, this section introduces two example problems, a harmonic oscillator and a mass-spring system. At the end of this section, numerical experiments for the DR algorithm and the AMPL–Ipopt suite and their comparisons are given for these two problems. Finally, concluding remarks and comments for future work are provided in Section 6.

## 2. Optimal control problem

In this, section we formulate the general optimal control problem that will be the focus of this paper. We give some necessary definitions and provide conditions for optimality from optimal control theory.

Before introducing the optimal control problem, we will give some standard definitions. Unless otherwise stated, all vectors are column vectors. Let  $\mathcal{L}^2([t_0, t_f]; \mathbb{R}^q)$  be the Banach space of Lebesgue measurable functions given by  $z : [t_0, t_f] \rightarrow \mathbb{R}^q$ , with a finite  $\mathcal{L}^2$  norm, namely,

$$\mathcal{L}^2([t_0, t_f]; \mathbb{R}^q) := \left\{ z : [t_0, t_f] \rightarrow \mathbb{R}^q \mid \|z\|_{\mathcal{L}^2} := \left( \int_{t_0}^{t_f} \|z(t)\|^2 dt \right)^{1/2} < \infty \right\},$$

where  $\|\cdot\|$  is the  $\ell_2$  norm in  $\mathbb{R}^q$ . Furthermore,  $\mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^q)$  is the Sobolev space of absolutely continuous functions, namely,

$$\mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^q) := \left\{ z \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^q) \mid \dot{z} := dz/dt \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^q) \right\},$$

endowed with the norm

$$\|z\|_{\mathcal{W}^{1,2}} := (\|z\|_{\mathcal{L}^2}^2 + \|\dot{z}\|_{\mathcal{L}^2}^2)^{1/2}.$$

With these definitions, we can now state the general LQ control problem as follows:

$$(P) \begin{cases} \min \frac{1}{2} \int_{t_0}^{t_f} (x(t)^T Q(t)x(t) + u(t)^T R(t)u(t)) dt, \\ \text{subject to } \dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0, \quad x(t_f) = x_f, \\ \underline{u} \leq u(t) \leq \bar{u}, \quad \underline{x} \leq x(t) \leq \bar{x}. \end{cases}$$

The state variable  $x \in \mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^n)$ , with  $x(t) := (x_1(t), \dots, x_n(t)) \in \mathbb{R}^n$ , and the control variable  $u \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m)$ , with  $u(t) := (u_1(t), \dots, u_m(t)) \in \mathbb{R}^m$ . The time-varying matrices  $A : [t_0, t_f] \rightarrow \mathbb{R}^{n \times n}$ ,  $B : [t_0, t_f] \rightarrow \mathbb{R}^{n \times m}$ ,  $Q : [t_0, t_f] \rightarrow \mathbb{R}^{n \times n}$ , and  $R : [t_0, t_f] \rightarrow \mathbb{R}^{m \times m}$  are continuous. For every  $t \in [t_0, t_f]$ , the matrices  $Q(t)$  and  $R(t)$  are symmetric and respectively positive semi-definite and positive definite. For clarity of argument, these matrices are assumed to be diagonal, namely,  $Q(t) := \text{diag}(q_1(t), \dots, q_n(t))$  and  $R(t) := \text{diag}(r_1(t), \dots, r_m(t))$ . These diagonality assumptions particularly simplify proximal mapping expressions that appear later. The initial and terminal states are given as  $x_0$  and  $x_f$ , respectively.

### 2.1. Optimality conditions

In this section, we state the maximum principle by using the direct adjoining approach from [21, Theorem 4.1]. We note that the authors of [21] designate Theorem 4.1 in their paper as an ‘‘informal theorem’’ since it has not been proved fully (for the case of pure state and mixed control and state constraints). However, they also point to a paper by Maurer as an exception where Theorem 4.1 has been proved for the case of pure state and pure control constraints, which is exactly our setting. We start by defining the extended Hamiltonian function  $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times [t_0, t_f] \rightarrow \mathbb{R}$  for Problem (P) as follows:

$$H(x(t), u(t), \lambda(t), \mu^1(t), \mu^2(t), t) := \frac{\lambda_0}{2} (x(t)^T Q(t)x(t) + u(t)^T R(t)u(t)) + \lambda(t)^T (A(t)x(t) + B(t)u(t)) + \mu^1(t)^T (x(t) - \bar{x}(t)) + \mu^2(t)^T (\underline{x} - x(t)),$$

where the multiplier  $\lambda_0 \geq 0$ , the adjoint variable vector  $\lambda : [t_0, t_f] \rightarrow \mathbb{R}^n$  with  $\lambda(t) := (\lambda_1(t), \dots, \lambda_n(t)) \in \mathbb{R}^n$ , and the state constraint multiplier vectors  $\mu^1, \mu^2 : [t_0, t_f] \rightarrow \mathbb{R}^n$  with  $\mu^i(t) := (\mu_1^i(t), \dots, \mu_n^i(t)) \in \mathbb{R}^n$ ,  $i = 1, 2$ . For brevity, we use the following notation:

$$H[t] := H(x(t), u(t), \lambda(t), \mu^1(t), \mu^2(t), t).$$

The adjoint variable vector is assumed to satisfy

$$\dot{\lambda}(t) := -H_x[t] = -Q(t)x(t) - A(t)^T \lambda(t) - \mu^1(t) + \mu^2(t), \quad (2.1)$$

for almost every  $t \in [t_0, t_f]$ , where  $H_x := \partial H / \partial x$ .

We assume that the problem we are solving is normal, so we set  $\lambda_0 = 1$ .

**Maximum Principle.** Suppose that the pair

$$(x, u) \in \mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^n) \times \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m)$$

is optimal for Problem (P). Then, there exists a piecewise absolutely continuous adjoint variable vector  $\lambda$ , as defined in (2.1), and piecewise continuous multipliers  $\mu^1, \mu^2 \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^n)$ , such that, for almost every  $t \in [t_0, t_f]$ ,

$$\begin{aligned} u_i(t) &= \operatorname{argmin}_{u_i \leq v_i \leq \bar{u}_i} H(x(t), u_1(t), \dots, v_i, \dots, u_m(t), \lambda(t), \mu^1(t), \mu^2(t), t) \\ &= \operatorname{argmin}_{u_i \leq v_i \leq \bar{u}_i} \frac{1}{2} r_i(t) v_i^2 + \lambda(t)^T b_i(t) v_i, \end{aligned} \quad (2.2)$$

for  $i = 1, \dots, m$ , where  $b_i(t)$  is the  $i$ th column of the matrix  $B(t)$  and  $r_i(t)$  is the  $i$ th diagonal element of  $R(t)$ . Moreover, the multipliers  $\mu^1(t), \mu^2(t)$  must satisfy the complementarity conditions given by

$$\mu_i^1(t) \geq 0, \quad \mu_i^1(t)(x_i(t) - \bar{x}_i(t)) = 0, \quad (2.3)$$

$$\mu_i^2(t) \geq 0, \quad \mu_i^2(t)(\underline{x}_i(t) - x_i(t)) = 0, \quad (2.4)$$

for all  $i = 1, \dots, n$ .

Suppose that  $\underline{u}_i = -\infty$ ,  $\bar{u}_i = \infty$ ,  $i = 1, \dots, m$ , i.e., the control vector is unconstrained. Then, (2.2) becomes

$$H_{u_i}[t] = 0,$$

i.e.,

$$r_i(t)u_i(t) + b_i(t)^T \lambda(t) = 0, \quad (2.5)$$

$i = 1, \dots, m$ . Then, (2.5) can be solved for  $u_i(t)$  as follows:

$$u_i(t) = -\frac{1}{r_i(t)} b_i(t)^T \lambda(t), \quad (2.6)$$

for  $i = 1, \dots, m$ ; or by using the matrices  $B(t)$  and  $R(t)$ :

$$u(t) = -[R(t)]^{-1} B(t)^T \lambda(t). \quad (2.7)$$

When we consider the constraints on  $u$ , one gets the following from (2.2):

$$u_j(t) = \begin{cases} \underline{u}_j, & \text{if } \frac{1}{r_j(t)} b_j(t)^T \lambda(t) \geq \underline{u}_j, \\ -\frac{1}{r_j(t)} b_j(t)^T \lambda(t), & \text{if } \underline{u}_j \leq -\frac{1}{r_j(t)} b_j(t)^T \lambda(t) \leq \bar{u}_j, \\ \bar{u}_j, & \text{if } \frac{1}{r_j(t)} b_j(t)^T \lambda(t) \leq \bar{u}_j, \end{cases} \quad (2.8)$$

for almost every  $t \in [t_0, t_f]$ ,  $j = 1, \dots, m$ .

### 3. Splitting and proximal mappings

In this section, we rewrite Problem (P) as the minimization of the sum of two convex functions  $f$  and  $g$ , and we give the proximal mappings for these functions in Theorems 1 and 2.

We split the constraints from (P) into two sets  $\mathcal{A}, \mathcal{B}$ , given as

$$\mathcal{A} := \{(x, u) \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^n) \times \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m) \mid \text{which solves}$$

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0, \quad x(t_f) = x_f, \text{ a.e. } t \in [t_0, t_f], \quad (3.1)$$

$$\mathcal{B} := \{(x, u) \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^n) \times \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m) \mid \underline{x}_i \leq x_i(t) \leq \bar{x}_i, \underline{u}_j \leq u_j(t) \leq \bar{u}_j, \\ \text{a.e. } t \in [t_0, t_f], i = 1, \dots, n, j = 1, \dots, m\}. \quad (3.2)$$

Despite previously defining  $x \in \mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^n)$  in our sets  $\mathcal{A}, \mathcal{B}$ , we let  $x \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^n)$  to simplify the calculation of the proximal mappings. By the definition of  $\mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^n)$ , if  $x \in \mathcal{A}$ , then  $x \in \mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^n)$ .

We assume that the control system  $\dot{x}(t) = A(t)x(t) + B(t)u(t)$  is controllable; in other words, the control system can be driven from any  $x_0$  to any other  $x_f$ ; for a precise definition of controllability and the tests for controllability, see [29]. Then, there exists a (possibly not unique)  $u(\cdot)$  such that, when this  $u(\cdot)$  is substituted, the boundary-value problem given in the expression for  $\mathcal{A}$  has a solution  $x(\cdot)$ . In other words,  $\mathcal{A} \neq \emptyset$ . Also, clearly,  $\mathcal{B} \neq \emptyset$ . We note that the constraint set  $\mathcal{A}$  is an affine subspace. Given that  $\mathcal{B}$  is a box, the constraints turn out to be two convex sets in Hilbert space. Since every sequence converging in  $\mathcal{L}^2$  has a subsequence converging pointwise almost everywhere, it is straightforward to see that the set  $\mathcal{B}$  is closed in  $\mathcal{L}^2$ . The closedness of  $\mathcal{A}$  will be established later as a consequence of Theorem 2 (see Remark 2).

Fix  $\beta > 0$  and let

$$f(x, u) := \iota_{\mathcal{B}}(x, u) + \frac{\beta}{2} \int_{t_0}^{t_f} (x(t)^T Q(t)x(t) + u(t)^T R(t)u(t)) dt \quad \text{and} \quad g(x, u) := \iota_{\mathcal{A}}(x, u), \quad (3.3)$$

where  $\iota_C$  is the indicator function of the set  $C$ , namely,

$$\iota_C(x, u) := \begin{cases} 0, & \text{if } (x, u) \in C, \\ \infty, & \text{otherwise.} \end{cases}$$

We note that the DE constraint in Problem (P) is represented by the indicator function  $\iota_{\mathcal{A}}(x, u)$  and the box constraints on  $(x, u)$  in Problem (P) are represented by the indicator function  $\iota_{\mathcal{B}}(x, u)$ . Problem (P) is then equivalent to the following problem:

$$\min_{x, u} f(x, u) + g(x, u). \quad (3.4)$$

The equivalence between Problem (P) and (3.4) follows from the fact that the objective function in (3.4) is the sum of the  $\beta$  weighted objective function of Problem (P) and the indicator functions of the constraints. Although the parameter  $\beta$  does not change the solution of the problem, it will have a role in the performance of the algorithm we study.

In our setting, we assume that we are able to compute the projector operators  $P_{\mathcal{A}}$  and  $P_{\mathcal{B}}$ . These operators project a given point onto each of the constraint sets  $\mathcal{A}$  and  $\mathcal{B}$ , respectively. Recall that the proximal mapping of a functional  $h$  is defined by [8, Definition 24.1]. For our setting,

$$\text{Prox}_h(x, u) := \underset{\substack{y \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^n) \\ v \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m)}}{\text{argmin}} \left( h(y, v) + \frac{1}{2} \|y - x\|_{\mathcal{L}^2}^2 + \frac{1}{2} \|v - u\|_{\mathcal{L}^2}^2 \right), \quad (3.5)$$

for any  $(x, u) \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^n) \times \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m)$ . Recall that the projection  $P_C(x, u)$  of a point  $(x, u)$  onto  $C$  is characterized by  $P_C(x, u) \in C$  and that  $\forall (y, v) \in C, \langle (y, v) - P_C(x, u), (x, u) - P_C(x, u) \rangle \leq 0$  [8, Theorem 3.16].

Note that  $\text{Prox}_{\iota_C} = P_C$ .

In order to implement the DR algorithm, we must write the proximal mappings  $f$  and  $g$ . The proofs of Theorems 1 and 2 below follow the broad lines of proof of Lemma 2 in [13]. In both theorems, the major difference from [13] is that the proximal operators in the current paper have two variables  $x^-$  and  $u^-$ . Thanks to separability, the proof of Theorem 1 is a straightforward modification of the corresponding part of the proof of [13, Lemma 2]. We include a full proof of Theorem 1 for the convenience of the reader. On the other hand, the proof of Theorem 2 deals with the solution of a more involved optimal control subproblem, namely, Problem (Pg).

**Theorem 1.** *The proximal mapping of  $f$  is given as  $\text{Prox}_f(x^-, u^-) = (y, v)$  such that the components of  $y$  and  $v$  are respectively expressed as follows*

$$y_i(t) = \begin{cases} \bar{x}_i, & \text{if } \frac{1}{\beta_{q_i+1}} x_i^-(t) \geq \bar{x}_i, \\ \frac{1}{\beta_{q_i+1}} x_i^-(t), & \text{if } \underline{x}_i \leq \frac{1}{\beta_{q_i+1}} x_i^-(t) \leq \bar{x}_i, \\ \underline{x}_i, & \text{if } \frac{1}{\beta_{q_i+1}} x_i^-(t) \leq \underline{x}_i, \end{cases} \quad (3.6)$$

$$v_j(t) = \begin{cases} \bar{u}_j, & \text{if } \frac{1}{\beta_{r_j+1}} u_j^-(t) \geq \bar{u}_j, \\ \frac{1}{\beta_{r_j+1}} u_j^-(t), & \text{if } \underline{u}_j \leq \frac{1}{\beta_{r_j+1}} u_j^-(t) \leq \bar{u}_j, \\ \underline{u}_j, & \text{if } \frac{1}{\beta_{r_j+1}} u_j^-(t) \leq \underline{u}_j, \end{cases} \quad (3.7)$$

for all  $t \in [t_0, t_f]$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ .

*Proof.* From (3.5) and the definition of  $f$  in (3.3), we have that

$$\begin{aligned} \text{Prox}_f(x^-, u^-) = \operatorname{argmin}_{x, u} & \iota_{\mathcal{B}}(x, u) + \frac{1}{2} \int_{t_0}^{t_f} \left( \beta x(t)^T Q(t) x(t) + \|x(t) - x^-(t)\|^2 \right. \\ & \left. + \beta u(t)^T R(t) u(t) + \|u(t) - u^-(t)\|^2 \right) dt. \end{aligned}$$

In other words, to find  $\text{Prox}_f(x^-, u^-)$  we need to find  $(y, v)$  that solves the following problem:

$$\text{(Pf)} \quad \begin{cases} \min & \frac{1}{2} \int_{t_0}^{t_f} \left( \beta y(t)^T Q(t) y(t) + \|y(t) - x^-(t)\|^2 + \beta v(t)^T R(t) v(t) + \|v(t) - u^-(t)\|^2 \right) dt, \\ \text{subject to} & \underline{x}_i \leq y_i(t) \leq \bar{x}_i, \quad \underline{u}_j \leq v_j(t) \leq \bar{u}_j \text{ for all } t \in [t_0, t_f], \quad i = 1, \dots, n, \quad j = 1, \dots, m. \end{cases}$$

Problem (Pf) is separable in the variables  $y$  and  $v$ , so we can consider the problems of minimizing with respect to  $y$  and  $v$  individually, and thus solve the following two subproblems:

$$\text{(Pf1)} \quad \begin{cases} \min & \frac{1}{2} \int_{t_0}^{t_f} \beta y(t)^T Q(t) y(t) + \|y(t) - x^-(t)\|^2 dt, \\ \text{subject to} & \underline{x}_i \leq y_i(t) \leq \bar{x}_i, \quad \text{for all } t \in [t_0, t_f], \quad i = 1, \dots, n, \end{cases}$$

and

$$\text{(Pf2)} \quad \begin{cases} \min & \frac{1}{2} \int_{t_0}^{t_f} \beta v(t)^T R(t) v(t) + \|v(t) - u^-(t)\|^2 dt, \\ \text{subject to} & \underline{u}_j \leq v_j(t) \leq \bar{u}_j \text{ for all } t \in [t_0, t_f], \quad j = 1, \dots, m. \end{cases}$$

The solution to Problem (Pf1) is given by

$$y_i(t) = \operatorname{argmin}_{x_i \leq z_i \leq \bar{x}_i} (\beta q_i z_i^2 + (z_i - x_i^-(t))^2),$$

$i = 1, \dots, n$ , which, after straightforward manipulations, yields (3.6). The solution to Problem (Pf2) is obtained similarly as:

$$v_j(t) = \operatorname{argmin}_{u_j \leq w_j \leq \bar{u}_j} (\beta r_j w_j^2 + (w_j - u_j^-(t))^2),$$

$j = 1, \dots, m$ , which yields (3.7) after straightforward manipulations.  $\square$

**Theorem 2.** *The proximal mapping of  $g$  is given as  $\operatorname{Prox}_g(x^-, u^-) = P_{\mathcal{A}}(x^-, u^-) = (y, v)$  such that*

$$y(t) = x(t), \quad (3.8)$$

$$v(t) = u^-(t) - B(t)^T \lambda(t), \quad (3.9)$$

where  $x(t), \lambda(t)$  are obtained by solving the following two-point boundary-value problem (TPBVP):

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u^-(t) - B(t)B(t)^T \lambda(t), & x(t_0) &= x_0, & x(t_f) &= x_f, \\ \dot{\lambda}(t) &= -x(t) + x^-(t) - A(t)^T \lambda(t). \end{aligned} \quad (3.10)$$

*Proof.* Using [8, Example 12.25], and the definition of  $g$  in (3.3),

$$\operatorname{Prox}_g(x^-, u^-) = \operatorname{Prox}_{\iota_{\mathcal{A}}}(x^-, u^-) = P_{\mathcal{A}}(x^-, u^-),$$

which verifies the very first assertion. From (3.5) and the definition of  $g$  in (3.3), we have that

$$\operatorname{Prox}_g(x^-, u^-) = \operatorname{argmin}_{x, u} \iota_{\mathcal{A}}(x, u) + \frac{1}{2} \int_{t_0}^{t_f} (\|x(t) - x^-(t)\|^2 + \|u(t) - u^-(t)\|^2) dt.$$

In other words, to find  $\operatorname{Prox}_g(x^-, u^-)$ , we need to find  $(y, v)$  that solves the following problem:

$$(Pg) \begin{cases} \min & \frac{1}{2} \int_{t_0}^{t_f} \|y(t) - x^-(t)\|^2 + \|v(t) - u^-(t)\|^2 dt, \\ \text{subject to} & \dot{y}(t) = A(t)y(t) + B(t)v(t), \quad y(t_0) = x_0, \quad y(t_f) = x_f. \end{cases}$$

Problem (Pg) is an optimal control problem wherein  $y(t)$  is the state variable and  $v(t)$  is the control variable. The Hamiltonian for Problem (Pg) is given by

$$H(y(t), v(t), \lambda(t), t) := \frac{1}{2} (\|y(t) - x^-(t)\|^2 + \|v(t) - u^-(t)\|^2) + \lambda(t)^T (A(t)y(t) + B(t)v(t)),$$

and the associated costate equation is given by

$$\dot{\lambda}(t) = -\partial H / \partial y = -y(t) + x^-(t) - A(t)^T \lambda(t). \quad (3.11)$$

If  $v$  is the optimal control for Problem (Pg), then, by the maximum principle,  $H_v[t] = 0$  for all  $t \in [t_0, t_f]$ . In other words,

$$v(t) - u^-(t) + B(t)^T \lambda(t) = 0,$$

for all  $t \in [t_0, t_f]$ , a re-arrangement of which yields (3.9). Collecting together the ODE in Problem (Pg) and the ODE in (3.11), substituting  $v(t)$  from (3.9), and assigning  $y(t) = x(t)$ , we can obtain the TPBVP in (3.10).  $\square$



**Remark 1.** We note from Theorem 2 that  $\text{Prox}_g$  is the projection onto the affine set  $\mathcal{A}$ . Unlike  $\text{Prox}_f$ , in general, we cannot find an analytical solution to (3.10) to obtain  $\text{Prox}_g$  (or the projection onto  $\mathcal{A}$ ) thus for this purpose, we will propose Algorithm 2 in Section 4.1 to get a numerical solution.

**Remark 2.** From Theorem 2, we see that every pair  $(x, u) \in \mathcal{L}^2 \times \mathcal{L}^2$  has a projection onto  $\mathcal{A}$ . In other words,  $\mathcal{A}$  is a Chebyshev set. It is well known that every Chebyshev set is closed (see [8, Remark 3.11(i)]). Hence, the set  $\mathcal{A}$  is closed in the topology of  $\mathcal{L}^2 \times \mathcal{L}^2$ .

#### 4. Douglas–Rachford algorithm

The application of the DR algorithm to our problem is slightly different from that in [7, 12]. Since we are solving control- and state-constrained optimal control problems, we must define the proximal mappings at the pair  $(x, u)$ , rather than just at  $u$ , as in [7, 12]. Thus, in the implementation of the DR algorithm, we give the iterations for the pair  $(x^k, u^k)$ , rather than for  $u^k$  alone.

Given  $\beta > 0$ , we specialize the DR algorithm (see [17, 18, 25]) to the case of minimizing the sum of the two functions  $f$  and  $g$  as in (3.3) and (3.4). The DR operator associated with the ordered pair  $(f, g)$  is defined by

$$T := \text{Id} - \text{Prox}_f + \text{Prox}_g(2 \text{Prox}_f - \text{Id}).$$

Application of the operator to our case is given by

$$T(x, u) = (x, u) - \text{Prox}_f(x, u) + P_{\mathcal{A}}(2 \text{Prox}_f(x, u) - (x, u)), \quad (4.1)$$

where the proximal mappings of  $f$  and  $g$  are provided as in Theorems 1 and 2. Let  $X$  be an arbitrary Hilbert space. Now, fix  $x_0 \in X$ . Given  $x_n \in X$ ,  $k \geq 0$ , the DR iterations are set as follows:

$$\begin{aligned} (b_{x,k}, b_{u,k}) &:= \text{Prox}_f(x_k, u_k), \\ (x_{k+1}, u_{k+1}) &:= T(x_k, u_k) = (x_k, u_k) - (b_{x,k}, b_{u,k}) + P_{\mathcal{A}}(2(b_{x,k}, b_{u,k}) - (x_k, u_k)). \end{aligned}$$

The DR algorithm is implemented as Algorithm 1. We define a new parameter  $\gamma := 1/(1 + \beta)$ , where  $\beta$  is the parameter that multiplies the objective as in (3.3) and Theorem 1. The choice of  $\gamma \in ]0, 1[$  can be made because changing  $\beta$  does not affect the solution of Problem (P).

---

#### Algorithm 1. (DR)

**Step 1. (Initialization)** Choose a parameter  $\gamma \in ]0, 1[$  and the initial iterate  $(x^0, u^0)$  arbitrarily. Choose a small parameter  $\varepsilon > 0$ , and set  $k = 0$ .

**Step 2. (Projection onto  $\mathcal{B}$ )** Set  $(x^-, u^-) = \gamma(x^k, u^k)$ . Compute  $(\bar{x}, \bar{u}) = P_{\mathcal{B}}(x^-, u^-)$  by using (3.6) and (3.7).

**Step 3. (Projection onto  $\mathcal{A}$ )** Set  $(x^-, u^-) := 2(\bar{x}, \bar{u}) - (x^k, u^k)$ . Compute  $(\widehat{x}, \widehat{u}) = P_{\mathcal{A}}(x^-, u^-)$  by using (3.8) and (3.9) or Algorithm 2.

**Step 4. (Update)** Set  $(x^{k+1}, u^{k+1}) := (x^k, u^k) + (\widehat{x}, \widehat{u}) - (\bar{x}, \bar{u})$ .

**Step 5. (Stopping criterion)** If  $\max(\|x^{k+1} - x^k\|_{L^\infty}, \|u^{k+1} - u^k\|_{L^\infty}) \leq \varepsilon$  or  $k \geq 200$ , then return  $(\bar{x}, \bar{u})$  and stop. Otherwise, set  $k := k + 1$  and go to Step 2.

---

**Remark 3.** We point out that, in general, only weak convergence is guaranteed for the DR algorithm (see [32, Theorem 1] or [10, Theorem 4.4]). It should be noted that the final iterate of the state-control pair in the box  $\mathcal{B}$  is returned by the algorithm.

#### 4.1. Algorithm for projector onto $\mathcal{A}$

We introduce a procedure for numerically projecting onto  $\mathcal{A}$ , that is, an extension of Algorithm 2 from [12] to the case of LQ control problems with state and control constraints. The procedure below (Algorithm 2) can be employed in Step 3 of the DR algorithm. In the procedure, we effectively solve the TPBVP in (3.10) by implementing the standard shooting method [4, 23, 31]. Throughout the steps of Algorithm 2 below, we will solve the ODEs in (3.10), rewritten here in matrix form as follows:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}^{DR}(t) \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)B(t)^T \\ -I_{n \times n} & -A(t)^T \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda^{DR}(t) \end{bmatrix} + \begin{bmatrix} 0_{n \times n} & B(t) \\ I_{n \times n} & 0_{n \times m} \end{bmatrix} \begin{bmatrix} x^-(t) \\ u^-(t) \end{bmatrix}, \quad (4.2)$$

with various initial conditions (ICs):

$$(i) \begin{bmatrix} x(t_0) \\ \lambda^{DR}(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}, \quad (ii) \begin{bmatrix} x(t_0) \\ \lambda^{DR}(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ e_i \end{bmatrix}, \quad (iii) \begin{bmatrix} x(t_0) \\ \lambda^{DR}(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \lambda_0^{DR} \end{bmatrix}. \quad (4.3)$$

In the above equations, we use  $\lambda^{DR}$ , instead of just  $\lambda$ , to emphasize the fact that  $\lambda^{DR}$  is the costate variable that results from solving Problem (Pg) to compute the projection onto  $\mathcal{A}$  within the DR algorithm. We reiterate that Problem (Pg) is more involved than its counterpart in [13], which leads to the ODE in (4.2), which, in turn, is more complicated than its counterpart in [12].

---

#### Algorithm 2. (Numerical Computation of the Projector onto $\mathcal{A}$ )

**Step 0. (Initialization)** The following parameters are given: Current iterate  $u^-$ , the system and control matrices  $A(t)$  and  $B(t)$ , the numbers of state and control variables  $n$  and  $m$ , and the initial and terminal states  $x_0$  and  $x_f$ , respectively. Define  $z(t, \lambda_0) := x(t)$ .

**Step 1. (Near-miss function)** Solve (4.2) with ICs in (4.3)(i) to find  $z(t_f, 0) = x(t_f)$ .  
Set  $\varphi(0) := z(t_f, 0) - x_f$ .

**Step 2. (Jacobian)** For  $i = 1, \dots, n$ , solve (4.2) with ICs in (4.3)(ii) to get  $z(t_f, e_i)$ .  
Set  $\beta_i(t) := z(t_f, e_i) - z(t_f, 0)$  and  $J_\varphi(0) := [\beta_1(t) \mid \dots \mid \beta_n(t)]$ .

**Step 3. (Missing IC)** Solve  $J_\varphi(0) \lambda_0^{DR} := -\varphi(0)$  for  $\lambda_0^{DR}$ .

**Step 4. (Projector onto  $\mathcal{A}$ )** Solve (4.2) with ICs in (4.3)(iii) to find  $x(t)$  and  $\lambda^{DR}(t)$ .  
Set  $P_{\mathcal{A}}(x^-, u^-)(t) := (y(t), v(t))$ , where  $y(t) = x(t)$  and  $v(t) = u^-(t) - B(t)^T \lambda^{DR}(t)$ .

---

#### 4.2. A conjecture for the costates for Problem (P)

Recall that the optimal control for Problem (P) is given by the cases in (2.8).

A junction time  $\bar{t}_j$  is a time when the control  $u_j(t)$  falls into two cases of (2.8) simultaneously, i.e., a point in time when a control constraint transitions from “active” to “inactive”, or vice versa. This definition of a junction time becomes important in the following conjecture, which has been formulated and tested by means of extensive numerical experiments.

**Conjecture 1.** Let  $\lambda^{DR}(t)$  be the costate variable emerging from the projection into  $\mathcal{A}$  as computed in Algorithm 2, and  $\lambda(t)$  be the costate variable that results from Problem (P). Let  $\bar{t}_j$  be a junction time for some  $u_j$ ,  $j = 1, \dots, m$ , such that  $b_j(\bar{t}_j)^T \lambda^{DR}(\bar{t}_j) \neq 0$ . Define

$$\alpha := \frac{-r_j(\bar{t}_j)u_j(\bar{t}_j)}{b_j(\bar{t}_j)^T \lambda^{DR}(\bar{t}_j)}.$$

Then,

$$\lambda(t) = \alpha \lambda^{DR}(t). \quad (4.4)$$

**Remark 4.** The ability to obtain the costate variable by Conjecture 1 is desirable as a tool for checking that the necessary condition of optimality in (2.8) is satisfied. Without this conjecture, we are unable to verify whether the optimality condition is satisfied when using the DR algorithm, except when a dual version of the DR algorithm is employed, as in [13].

Once we have calculated  $\lambda$  in this way, we can also find a multiplier  $\mu^k$ ,  $k = 1, 2$ , numerically, for the case in which only one state constraint is active at a given time. Suppose that only the  $i$ th state box constraint becomes active. By rearranging (2.1), using numerical differentiation to find  $\dot{\lambda}$ , and assuming  $\mu_i^2(t) = 0$ , we have that

$$\mu_i^1(t) = -Q(t)x(t) - A(t)^T \lambda(t) - \dot{\lambda}(t). \quad (4.5)$$

If  $\mu_i^1(t) = 0$ , then we compute the following:

$$\mu_i^2(t) = Q(t)x(t) + A(t)^T \lambda(t) + \dot{\lambda}(t). \quad (4.6)$$

With (4.5), or with (4.6), the complementarity conditions in (2.3) or (2.4) can now be checked numerically.

## 5. Numerical experiments

We will now introduce two example problems. Along with posing the optimal control problems, we also give plots of their optimal controls, states, costates, and multipliers, with vertical lines signifying the regions in which the state constraints become active.

### 5.1. Harmonic oscillator

Problem (PHO) below contains the dynamics of a harmonic oscillator which, is typically used to model a point mass with a spring. The dynamics are given as  $\ddot{y}(t) + \omega_0^2 y(t) = f(t)$ , where  $\omega_0 > 0$  is known as the natural frequency and  $f(t)$  is some forcing. In a physical system,  $y$  represents the position of a unit mass,  $\dot{y}$  is the velocity of said mass, the natural frequency is defined as  $\omega_0 = \sqrt{k}$ , where  $k$  is the stiffness of the spring producing the harmonic motion, and  $f$  is the restoring force. In addition to the restoring force, we will introduce another force  $u_1$  that will affect the velocity  $\dot{y}$  directly. We let  $x_1 := y$ ,  $x_2 := \dot{y}$ , and  $u_2 := f$  to arrive at  $\dot{x}_1(t) = x_2(t) + u_1(t)$ ,  $\dot{x}_2(t) = -\omega_0^2 x_1(t) + u_2(t)$ .

In this example problem, the objective contains the squared sum of all four variables in the system. It is common to consider this problem with the objective of minimizing the energy of the control

variable, but, in this case, we have also included the state variables to test the algorithm with a slightly more involved objective. The focus of this research is control- and state-constrained problems, so the constraints have been added as in Problem (P):

$$(PHO) \left\{ \begin{array}{l} \min \quad \frac{1}{2} \int_0^{2\pi} x_1^2(t) + x_2^2(t) + u_1^2(t) + u_2^2(t) dt, \\ \text{subject to} \quad \dot{x}_1(t) = x_2(t) + u_1(t), \quad x_1(0) = 0, \quad x_2(2\pi) = 0, \\ \quad \quad \quad \dot{x}_2(t) = -4x_1(t) + u_2(t), \quad x_2(0) = 1, \quad x(2\pi) = 0, \\ \quad \quad \quad \underline{u} \leq u(t) \leq \bar{u}, \quad \underline{x} \leq x(t) \leq \bar{x}. \end{array} \right.$$

### 5.2. Simple spring-mass system

The simple spring-mass system is another physical system that can be easily visualized, see [30]. This problem contains two masses and two springs that are connected in sequence with the dynamics given by  $m_1\ddot{y}_1(t) + (k_1 + k_2)y_1(t) - k_2y_2 = f_1(t)$ ,  $m_2\ddot{y}_2(t) - k_2y_1(t) + k_2y_2(t) = f_2(t)$ , where  $m_1, m_2$  are the two masses,  $k_1, k_2$  are the spring coefficients (stiffness), and  $f_1(t), f_2(t)$  are the forces applied to  $m_1, m_2$ , respectively. Let  $x_1 := y_1$ ,  $x_2 := \dot{y}_1$ ,  $x_3 := y_2$ ,  $x_4 := \dot{y}_2$ ,  $u_1 := f_1$ ,  $u_2 := f_2$ ,  $m_1 = m_2 = 1$ ,  $k_1 = 1$ , and  $k_2 = 2$ ; then, we retrieve the system in Problem (PSM). This dynamical system furnishes an optimal control problem with four state variables and two control variables. As in (PHO), we have added state and control constraints and set the objective as the integral of the squared sum of the state and control variables:

$$(PSM) \left\{ \begin{array}{l} \min \quad \frac{1}{2} \int_0^{2\pi} x_1^2(t) + x_2^2(t) + x_3^2(t) + x_4^2(t) + u_1^2(t) + u_2^2(t) dt, \\ \text{subject to} \quad \dot{x}_1(t) = x_2(t), \quad x_1(0) = 0, \quad x_1(2\pi) = 0, \\ \quad \quad \quad \dot{x}_2(t) = -3x_1(t) + 2x_3(t) + u_1(t), \quad x_2(0) = 1, \quad x_2(2\pi) = 0, \\ \quad \quad \quad \dot{x}_3(t) = x_4(t), \quad x_3(0) = 1, \quad x_3(2\pi) = 0, \\ \quad \quad \quad \dot{x}_4(t) = 2x_1(t) - 2x_3(t) + u_2(t), \quad x_4(0) = -1, \quad x_4(2\pi) = 0, \\ \quad \quad \quad \underline{u} \leq u(t) \leq \bar{u}, \quad \underline{x} \leq x(t) \leq \bar{x}. \end{array} \right.$$

### 5.3. Numerical discussion and comparisons

**Technical specifications.** In the numerical experiments, we used MATLAB version 2023b with the DR algorithm and compared its performance with that of the AMPL–Ipopt optimization computational suite [19, 33] with Ipopt version 3.12.13. We chose to make comparisons to Ipopt since it is a free and easily obtainable solver that is used for problems such as those presented in this paper (also see [5]). All computations applied  $\epsilon = 10^{-8}$  from Algorithm 1 or, in the case of Ipopt,  $\text{tol} = 10^{-8}$ , and were run on a PC with an i5-10500T 2.30GHz processor with 8GB RAM. For the two examples (PHO) and (PSM), we experimented with a case with only control constraints and a case in which there was also an added state constraint. The problem specifications can be found in Table 1, along with the choices of  $\gamma$  that were used in the implementation of the DR algorithm.

**Table 1.** Problem cases.

Problem	Case	$\gamma$	Constraints
(PHO)	Case 1	0.60	$-0.4 \leq u_1(t) \leq 0.1, -0.5 \leq u_2(t) \leq 0.1$
	Case 2	0.95	$-0.4 \leq u_1(t) \leq 0.1, -0.5 \leq u_2(t) \leq 0.1, -0.025 \leq x_1(t)$
(PSM)	Case 1	0.55	$-0.5 \leq u_1(t) \leq 0.5, -0.4 \leq u_2(t) \leq 0.4$
	Case 2	0.95	$-0.5 \leq u_1(t) \leq 0.5, -0.4 \leq u_2(t) \leq 0.4, -0.2 \leq x_1(t)$

**Generation of “true” solutions.** Since we did not have analytical solutions to these problems, we generated higher-accuracy numerical solutions to our problems that we will use as “true” solutions in our error calculations. In the Case 1 examples (only control constraints present) the DR algorithm was able to successfully converge to an acceptable solution by using  $N = 10^7$  and  $\epsilon = 10^{-12}$ , without reaching 200 iterations (the maximum number of iterations we allowed). The 8 GB of RAM provided insufficient memory for Ipopt to generate a solution with such a high number of discretization points, so the DR algorithm was used to generate the “true” solutions in this case. In Case 2 (state-constrained case), the DR algorithm was unable to converge to an accepted solution in less than 200 iterations, so we instead relied on Ipopt. Due to the memory limitations, the “true” solution for (PHO) Case 2 was generated with  $N = 10^6$ ,  $\epsilon = 10^{-12}$ , and (PSM) Case 2 used  $N = 7 \times 10^5$ ,  $\epsilon = 10^{-12}$ .

**Choice of an algorithmic parameter.** The values of  $\gamma$  in Table 1 were decided by generating plots that display the number of iterations required for the DR algorithm to find an acceptable solution for 500 values of  $\gamma$  in the interval  $(0, 1)$ . For both (PHO) and (PSM) in Case 2, for all values of  $\gamma$  that were tested the DR algorithm required more than the maximum 200 iterations to converge. For these cases, we instead generated plots that compared the errors in the states and controls for the different values of  $\gamma$ . A specific value of  $\gamma$  that would provide the smallest errors was not obvious since many values provided similar performance; but, values closer to 1 appeared optimal; thus, the choice of  $\gamma = 0.95$  was made for these experiments.

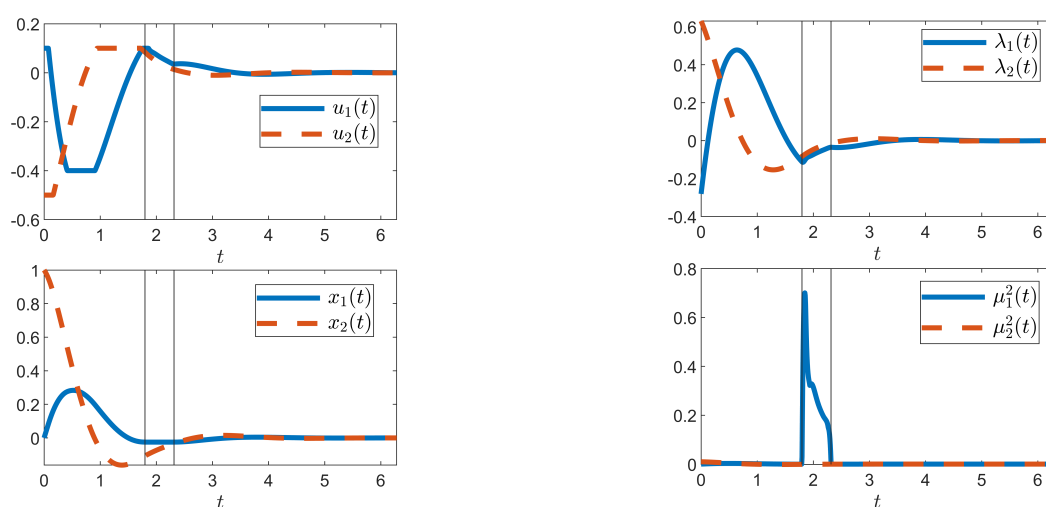
**Graphical results.** In Figures 1 and 2, we have plots for (PHO) and (PSM) using the DR algorithm. We have generated similar figures based on the results of Ipopt as well, but, since they mostly overlap Figures 1 and 2, we will not show them all here but will point out the differences. In Figure 3, we give the multiplier vector  $\mu^2$  components for Case 2 (PHO) as obtained by using the DR algorithm and Ipopt. As indicated by the black vertical lines in the bottom left plot of Figure 1, when using the DR algorithm the constraint on  $x_1$  is active over a time interval that aligns with the interval in which  $\mu_1^2$  is positive, as expected from (2.3) and (2.4).

In the results from Ipopt, we observe that the state constraint only became active at a single point in time, but Figure 3 shows that  $\mu_1^2$  from Ipopt (yellow dotted line) is positive for a larger interval of time; thus, (2.4) is violated. This appears to be a numerical error that is not present in the DR results since there is an interval of points around the spike where the state constraint  $-0.025 \leq x_1(t)$  becomes active. When  $N = 10^4, 10^5$ , for (PHO), we observe that, when using Ipopt, the lower bound on  $x_1$  is never reached, though, again, we see the interval of points that are almost equal to the lower bound.

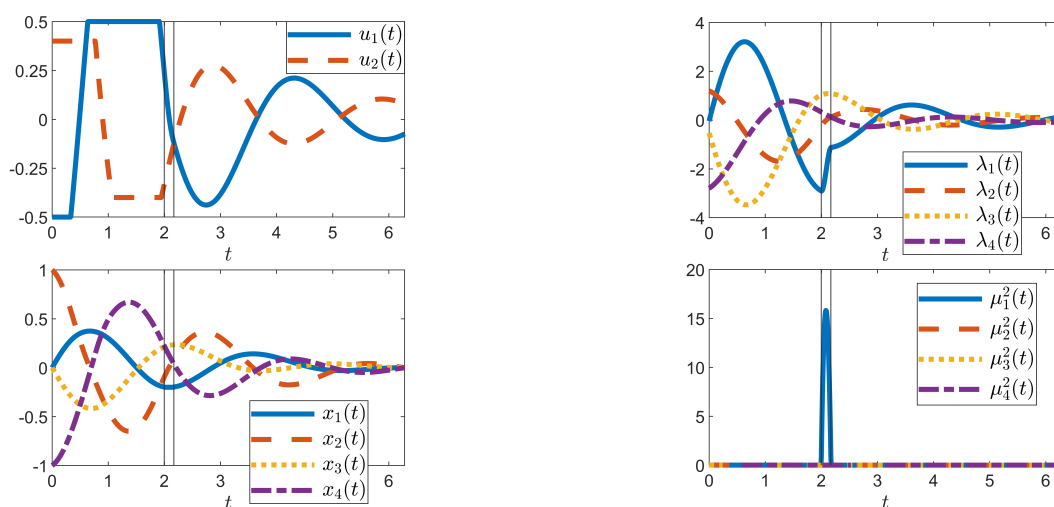
In Figure 3, we also see a discrepancy in  $\mu_2^2$  since we have not imposed a constraint on the variable  $x_2$ . Equation (2.4) implies that  $\mu_2^2(t) = 0$  for all  $t \in [t_0, t_f]$ . We can see in Figure 3 that  $\mu_2^2$  from Ipopt (purple dotted line) fails to satisfy this requirement, while  $\mu_2^2$  from the DR algorithm is, at least to the eye, equal to zero.

Another difference between the multipliers  $\mu$  from the DR algorithm and Ipopt can be seen in

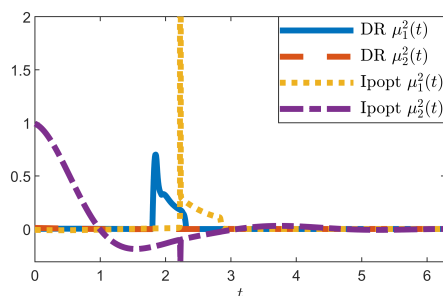
the maximum values reached by the functions. In Figure 3, we see that  $\mu_1^2$  obtained via the DR algorithm and Ipopt have similar shapes in their plots, but the maximum value reached by using the DR algorithm is approximately 0.7, while, from Ipopt, the maximum value is approximately 33. For (PSM) Case 2 with  $N = 10^3$ , the maximum value for  $\mu_1^2$  that was obtained by using the DR algorithm was approximately 16, while Ipopt yielded a maximum value of approximately 310. Along with the functions having very different maximum values, we noted that, when generating these plots for  $N = 10^3, 10^4, 10^5$ , the results from the DR algorithm were clearly converging to a single function  $\mu_1^2$ , while this was not obvious from the Ipopt results. For (PSM), we see the approximate maximum values of  $\mu_1^2$  that were obtained when using Ipopt were 310 for  $N = 10^3$ , 2500 for  $N = 10^4$ , and 3000 for  $N = 10^5$ . In the same example, using the DR algorithm, the maximum values of the function were 16 for  $N = 10^3$ , 14 for  $N = 10^4$ , and 14 for  $N = 10^5$ .



**Figure 1.** (PHO) Case 2 plots (see Table 1) using the DR algorithm with  $N = 10^3$ ,  $-0.025 \leq x_1(t)$ . Vertical lines indicate the interval in which the state constraint becomes active.



**Figure 2.** (PSM) Case 2 plots (see Table 1) using the DR algorithm with  $N = 10^3$ ,  $-0.2 \leq x_1(t)$ . Vertical lines indicate the interval in which the state constraint becomes active.



**Figure 3.** Multipliers  $\mu_1^2, \mu_2^2$  for Case 2 (PHO) with  $N = 10^3$ ; as obtained by using the DR algorithm (solid lines) and Ipopt (dotted lines). Note that  $\mu_1^2(t)$  found by Ipopt attained a maximum value of 33, which is not shown in the graph.

We also observed some slight variation between the plots obtained using the DR algorithm and Ipopt at the junction times in the control variables (not shown in this paper in order to avoid excessive amount of visual material). The intervals in which the control variables reached their boundaries via the DR algorithm always appeared slightly larger than those obtained from Ipopt. The control variables in the region in which they transition between active and inactive constraints appeared more rounded at these corners when using Ipopt and exhibited a sharper transition when using the DR algorithm.

**Errors and CPU times.** Table 2 contains the errors in the controls, states, and costates for the DR algorithm and Ipopt, while Table 3 contains the errors in the multipliers, objective values, and CPU times. The CPU times were computed as averages of over 200 runs (up to 1,000 runs in the faster examples). The values within boxes are the smaller errors and CPU times between the DR algorithm and Ipopt. At a glance, we can see that, more often than not, the DR algorithm produced smaller errors and faster CPU times than Ipopt. Upon closer inspection, we see that, in many of the Case 2 results, the errors from the DR algorithm and Ipopt are comparable. We note that the “true” solutions used to calculate these errors are those explained earlier in this subsection, except for the multipliers  $\mu^2$  in the Case 1 examples. In those examples, we applied the zero vector as our “true” solution.

We observe that most of the results in Table 2 show a smaller error in the control variable from the DR algorithm, especially in the Case 1 examples where there are no state constraints. Regarding the state variables, we see that Ipopt has smaller errors when  $N = 10^3, 10^4$ , though there is little difference from the DR algorithm results, and we see an improvement in the DR algorithm when  $N = 10^5$ . Like with the error in the control variables, we see that the error in the costates is smaller for the DR algorithm in almost all examples.

From Table 3, the errors in the multipliers show an improvement in the DR results relative to Ipopt, though the “true” solution in the Case 2 examples was obtained by using results from Ipopt, which, as previously mentioned, did not appear to converge to a specific value as we increased  $N$  thus, the quality of this “true” solution is not guaranteed. The DR algorithm produced slightly smaller objective values that were closest to the “true” solution in almost all cases, though the difference compared with Ipopt was marginal. We see that the CPU times were faster for the DR algorithm, especially in the examples where  $N = 10^5$ .

**Table 2.** Errors in controls  $u$ , states  $x$ , and costates  $\lambda$  and CPU times for the DR algorithm and AMPL–Ipopt, with  $\varepsilon = 10^{-8}$  and specifications from Table 1.

$N$	Problem	$L^\infty$ error in control		$L^\infty$ error in states		$L^\infty$ error in costates	
		DR	Ipopt	DR	Ipopt	DR	Ipopt
$10^3$	(PHO)-Case 1	$7.9 \times 10^{-3}$	$1.4 \times 10^{-2}$	$6.9 \times 10^{-3}$	$2.7 \times 10^{-3}$	$6.3 \times 10^{-3}$	$1.4 \times 10^{-2}$
	(PHO)-Case 2	$1.4 \times 10^{-2}$	$1.6 \times 10^{-2}$	$4.5 \times 10^{-3}$	$2.9 \times 10^{-3}$	$1.4 \times 10^{-2}$	$1.6 \times 10^{-2}$
	(PSM)-Case 1	$2.3 \times 10^{-2}$	$3.8 \times 10^{-2}$	$2.0 \times 10^{-2}$	$1.8 \times 10^{-2}$	$5.5 \times 10^{-2}$	$1.5 \times 10^{-1}$
	(PSM)-Case 2	$7.1 \times 10^{-2}$	$1.1 \times 10^{-1}$	$3.8 \times 10^{-1}$	$3.7 \times 10^{-1}$	$8.4 \times 10^{-1}$	$1.5 \times 10^0$
$10^4$	(PHO)-Case 1	$7.8 \times 10^{-4}$	$3.4 \times 10^{-3}$	$6.7 \times 10^{-4}$	$3.6 \times 10^{-4}$	$7.5 \times 10^{-4}$	$1.6 \times 10^{-3}$
	(PHO)-Case 2	$1.3 \times 10^{-3}$	$3.0 \times 10^{-3}$	$4.7 \times 10^{-4}$	$4.2 \times 10^{-4}$	$5.0 \times 10^{-3}$	$4.8 \times 10^{-3}$
	(PSM)-Case 1	$2.2 \times 10^{-3}$	$3.6 \times 10^{-3}$	$2.0 \times 10^{-3}$	$1.8 \times 10^{-3}$	$4.8 \times 10^{-3}$	$1.5 \times 10^{-2}$
	(PSM)-Case 2	$2.4 \times 10^{-2}$	$1.1 \times 10^{-2}$	$3.7 \times 10^{-1}$	$3.7 \times 10^{-1}$	$7.9 \times 10^{-1}$	$1.5 \times 10^0$
$10^5$	(PHO)-Case 1	$7.7 \times 10^{-5}$	$6.3 \times 10^{-3}$	$6.7 \times 10^{-5}$	$9.5 \times 10^{-4}$	$6.5 \times 10^{-5}$	$2.1 \times 10^{-3}$
	(PHO)-Case 2	$7.8 \times 10^{-4}$	$7.5 \times 10^{-3}$	$6.4 \times 10^{-5}$	$1.4 \times 10^{-3}$	$5.7 \times 10^{-3}$	$7.3 \times 10^{-3}$
	(PSM)-Case 1	$2.2 \times 10^{-4}$	$1.1 \times 10^{-2}$	$2.0 \times 10^{-4}$	$6.8 \times 10^{-4}$	$4.3 \times 10^{-4}$	$4.0 \times 10^{-3}$
	(PSM)-Case 2	$2.6 \times 10^{-2}$	$6.0 \times 10^{-3}$	$3.7 \times 10^{-1}$	$3.7 \times 10^{-1}$	$8.0 \times 10^{-1}$	$7.9 \times 10^{-1}$

**Table 3.** Errors in multipliers  $\mu$ , errors in objective values, and CPU times for the DR algorithm and AMPL–Ipopt, with  $\varepsilon = 10^{-8}$  and specifications from Table 1.

$N$	Problem	$L^\infty$ error in $\mu$		$L^\infty$ error in objective values		CPU time [s]	
		DR	Ipopt	DR	Ipopt	DR	Ipopt
$10^3$	(PHO)-Case 1	$4.9 \times 10^{-3}$	$3.1 \times 10^{-2}$	$2.9 \times 10^{-3}$	$4.8 \times 10^{-3}$	$4.8 \times 10^{-2}$	$2.9 \times 10^{-1}$
	(PHO)-Case 2	$5.4 \times 10^{-1}$	$1.5 \times 10^0$	$2.9 \times 10^{-3}$	$4.9 \times 10^{-3}$	$3.3 \times 10^{-1}$	$3.7 \times 10^{-1}$
	(PSM)-Case 1	$2.4 \times 10^{-3}$	$8.6 \times 10^{-2}$	$4.8 \times 10^{-2}$	$5.1 \times 10^{-2}$	$1.0 \times 10^{-1}$	$4.4 \times 10^{-1}$
	(PSM)-Case 2	$1.6 \times 10^1$	$3.1 \times 10^0$	$6.8 \times 10^{-2}$	$7.4 \times 10^{-2}$	$5.2 \times 10^{-1}$	$5.3 \times 10^{-1}$
$10^4$	(PHO)-Case 1	$1.5 \times 10^{-4}$	$3.0 \times 10^{-3}$	$2.8 \times 10^{-4}$	$4.8 \times 10^{-4}$	$4.7 \times 10^{-1}$	$2.7 \times 10^0$
	(PHO)-Case 2	$1.7 \times 10^1$	$1.7 \times 10^1$	$2.8 \times 10^{-4}$	$4.9 \times 10^{-4}$	$3.5 \times 10^0$	$3.3 \times 10^0$
	(PSM)-Case 1	$7.1 \times 10^{-4}$	$8.2 \times 10^{-3}$	$4.6 \times 10^{-3}$	$5.0 \times 10^{-3}$	$1.1 \times 10^0$	$4.6 \times 10^0$
	(PSM)-Case 2	$1.4 \times 10^1$	$2.5 \times 10^0$	$4.4 \times 10^{-3}$	$6.9 \times 10^{-3}$	$5.9 \times 10^0$	$6.4 \times 10^0$
$10^5$	(PHO)-Case 1	$3.7 \times 10^{-5}$	$3.0 \times 10^{-4}$	$2.8 \times 10^{-5}$	$7.7 \times 10^{-5}$	$5.1 \times 10^0$	$2.6 \times 10^1$
	(PHO)-Case 2	$1.8 \times 10^1$	$1.8 \times 10^1$	$2.4 \times 10^{-5}$	$1.4 \times 10^{-4}$	$3.7 \times 10^1$	$3.1 \times 10^1$
	(PSM)-Case 1	$1.1 \times 10^{-4}$	$8.2 \times 10^{-4}$	$4.5 \times 10^{-4}$	$5.3 \times 10^{-4}$	$1.2 \times 10^1$	$1.2 \times 10^2$
	(PSM)-Case 2	$2.2 \times 10^2$	$3.7 \times 10^3$	$1.5 \times 10^{-3}$	$7.2 \times 10^{-4}$	$6.6 \times 10^1$	$1.6 \times 10^2$



## 6. Conclusions

We have applied the DR algorithm to find a numerical solution of LQ control problems with state and control constraints after re-formulating these problems as the sum of two convex functions and deriving expressions for the proximal mappings of these functions (Theorems 1 and 2). These proximal mappings were used in the DR iterations. Within the DR algorithm (Algorithm 1), we proposed a procedure (Algorithm 2) for finding the projection onto the affine set defined by the ODEs numerically.

We carried out extensive numerical experiments on two nontrivial example problems and illustrated both the performance of the algorithm and its efficiency (in both accuracy and speed) compared to the traditional approach of direct discretization. We observed that, in general, the DR algorithm produced smaller errors and faster run times for these problems, most notably in the examples where we have increased the number of discretization points. From these numerical results, the DR algorithm could, in general, be recommended over Ipopt when high-quality solutions are desired.

Based on further extensive experiments, we conjectured on how the costate variables can be determined. We successfully used the costate variables constructed as in the conjecture, as well as the state constraint multipliers that can be calculated by using these costate variables, for the numerical verification of the optimality conditions.

We recall that Algorithm 2 involves repeated numerical solution of the ODEs in (4.2) with various ICs. To solve (4.2), we implemented the (explicit) Euler's method, which requires only a continuous right-hand side of the ODEs. Algorithm 2 appears to be successful for the examples partly owing to the fact that, in these examples, the optimal control is continuous. We tried to apply our approach to the machine tool manipulator problem from [16], which has 7 state variables, one control variable, and upper and lower bounds imposed on the single control variable and one of the state variables. However, our approach did not seem to yield a solution (so far) for this problem, conceivably owing to the fact that the optimal control variable, as well as the optimal costate variable vector, is not continuous, in that these variables jump a number of times during the process (see [16, Figure 5]). Note that discontinuities in the control make the right-hand side of (4.2) discontinuous, rendering Euler's method ineffective. Therefore, problems of the kind in [16] require further investigation.

We believe that our approach can be extended to more general convex optimal control problems, such as those with a non-quadratic objective function or mixed state and control constraints, as long as the pertaining proximal operators are not too challenging to derive.

It would also be interesting to employ and test, in the future, other projection-type methods, such as the Peaceman-Rachford algorithm [8, Section 26.4 and Proposition 28.8], which, to the best of the knowledge of the authors, has not been applied to optimal control problems.

### Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

### Acknowledgments

The authors thank the two anonymous reviewers whose comments and suggestions improved the paper. BIC was supported by an Australian government research training program scholarship.

## Conflict of interest

The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. H. M. Amman, D. A. Kendrick, Computing the steady state of linear quadratic optimization models with rational expectations, *Econ. Lett.*, **58** (1998), 185–191. [http://dx.doi.org/10.1016/S0165-1765\(97\)00263-2](http://dx.doi.org/10.1016/S0165-1765(97)00263-2)
2. F. J. Aragón Artacho, J. M. Borwein, M. K. Tam, Douglas–Rachford feasibility methods For matrix completion problems, *ANZIAM J.*, **55** (2014), 299–326. <http://dx.doi.org/10.1017/S1446181114000145>
3. F. J. Aragón Artacho, R. Campoy, V. Elser, An enhanced formulation for solving graph coloring problems with the Douglas–Rachford algorithm, *J. Glob. Optim.*, **77** (2020), 383–403. <http://dx.doi.org/10.1007/s10898-019-00867-x>
4. U. M. Ascher, R. M. M. Mattheij, R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Philadelphia: SIAM Publications, 1995. <http://dx.doi.org/10.1137/1.9781611971231>
5. N. Banihashemi, C. Y. Kaya, Inexact restoration for Euler discretization of box-constrained optimal control problems, *J. Optim. Theory Appl.*, **156** (2013), 726–760. <http://dx.doi.org/10.1007/s10957-012-0140-4>
6. H. H. Bauschke, *8 queens, sudoku, and projection methods*, The Mathematical Interests of Peter Borwein, The IRMACS Centre, 2008. Available from: <https://carmamaths.org/resources/jon/Preprints/Talks/InverseProbs/Earlier/Bauschke-IRMACS08.pdf>.
7. H. H. Bauschke, R. S. Burachik, C. Y. Kaya, Constraint splitting and projection methods for optimal control of double integrator, In: *Splitting Algorithms, Modern Operator Theory, and Applications*, Cham: Springer, 45–68, 2019. [http://dx.doi.org/10.1007/978-3-030-25939-6\\_2](http://dx.doi.org/10.1007/978-3-030-25939-6_2)
8. H. H. Bauschke, P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Cham: Springer, 2017. <http://dx.doi.org/10.1007/978-3-319-48311-5>
9. H. H. Bauschke, V. R. Koch, Projection methods: Swiss Army knives for solving feasibility and best approximation problems with halfspaces, arXiv: 1301.4506.
10. H. H. Bauschke, W. M. Moursi, On the Douglas–Rachford algorithm, *Math. Program.*, **164** (2017), 263–284. <http://dx.doi.org/10.1007/s10107-016-1086-3>
11. R. S. Burachik, B. I. Caldwell, C. Y. Kaya, Projection methods for control-constrained minimum-energy control problems, arXiv: 2210.17279.
12. R. S. Burachik, B. I. Caldwell, C. Y. Kaya, Douglas–Rachford algorithm for control-constrained minimum-energy control problems, *ESAIM: COCV*, **30** (2024), 18 <http://dx.doi.org/10.1051/cocv/2024004>

13. R. S. Burachik, B. I. Caldwell, C. Y. Kaya, W. M. Moursi, Optimal control duality and the Douglas–Rachford algorithm, *SIAM J. Control Optim.*, **62** (2024), 680–698. <http://dx.doi.org/10.1137/23M1558549>
14. R. S. Burachik, C. Y. Kaya, S. N. Majeed, A duality approach for solving control-constrained linear-quadratic optimal control problems, *SIAM J. Control Optim.*, **52** (2014), 1423–1456. <http://dx.doi.org/10.1137/130910221>
15. C. Büskens, H. Maurer, SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control, *J. Comput. Appl. Math.*, **120** (2000), 85–108. [http://dx.doi.org/10.1016/S0377-0427\(00\)00305-8](http://dx.doi.org/10.1016/S0377-0427(00)00305-8)
16. B. Christiansen, H. Maurer, O. Zirn, Optimal control of machine tool manipulators, In: *Recent Advances in Optimization and Its Applications in Engineering*, Berlin: Springer, 2010, 451–460. [http://dx.doi.org/10.1007/978-3-642-12598-0\\_39](http://dx.doi.org/10.1007/978-3-642-12598-0_39)
17. J. Douglas, H. H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, *Trans. Amer. Math. Soc.*, **82** (1956), 421–439. <http://dx.doi.org/10.2307/1993056>
18. J. Eckstein, D. P. Bertsekas, On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Math. Program.*, **55** (1992), 293–318. <http://dx.doi.org/10.1007/BF01581204>
19. R. Fourer, D. M. Gay, B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, 2 Eds., New York: Duxbury/Thomson, 2003. Available from: <https://ampl.com/wp-content/uploads/BOOK.pdf>.
20. S. Gravel, V. Elser, Divide and concur: a general approach to constraint satisfaction, *Phys. Rev. E*, **78** (2008), 036706. <http://dx.doi.org/10.1103/PhysRevE.78.036706>
21. R. F. Hartl, S. P. Sethi, R. G. Vickson, A survey of the maximum principles for optimal control problems with state constraints, *SIAM Rev.*, **37** (1995), 181–218. <http://dx.doi.org/10.1137/1037043>
22. C. Y. Kaya, Inexact restoration for Runge–Kutta discretization of optimal control problems, *SIAM J. Numer. Anal.*, **48** (2010), 1492–1517. <http://dx.doi.org/10.1137/090766668>
23. H. B. Keller, *Numerical Methods for Two-point Boundary-Value Problems*, Philadelphia: Society for Industrial and Applied Mathematics, 1972.
24. B. Kugelmann, H. J. Pesch, New general guidance method in constrained optimal control, part 1: numerical method, *J. Optim. Theory Appl.*, **67** (1990), 421–435. <http://dx.doi.org/10.1007/BF00939642>
25. P.-L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.*, **16** (1979), 964–979. <http://dx.doi.org/10.1137/0716071>
26. H. Maurer, J.-H. R. Kim, G. Vossen, On a state-constrained control problem in optimal production and maintenance, In: *Optimal Control and Dynamic Games*, Boston: Springer Verlag, 2005, 289–308. [http://dx.doi.org/10.1007/0-387-25805-1\\_17](http://dx.doi.org/10.1007/0-387-25805-1_17)
27. H. Maurer, H. J. Oberle, Second order sufficient conditions for optimal control problems with free final time: the Riccati approach, *SIAM J. Control Optim.*, **41** (2002), 380–403. <http://dx.doi.org/10.1137/S0363012900377419>

28. T. Mouktonglang, Innate immune response via perturbed LQ-control problem, *Advanced Studies in Biology*, **3** (2011), 327–332.
29. W. J. Rugh, *Linear system theory*, 2 Eds., London: Pearson, 1996.
30. T. L. Schmitz, K. S. Smith, *Mechanical vibrations modeling and measurement*, New York: Springer, 2011. <http://dx.doi.org/10.1007/978-1-4614-0460-6>
31. J. Stoer, R. Bulirsch, *Introduction to numerical analysis*, 3 Eds., New York: Springer-Verlag, 2002. <http://dx.doi.org/10.1007/978-0-387-21738-3>
32. B. F. Svaiter, On weak convergence of the Douglas–Rachford method, *SIAM J. Control Optim.*, **49** (2011), 280–287. <http://dx.doi.org/10.1137/100788100>
33. A. Wächter, L. T. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming, *Math. Program.*, **106** (2006), 25–57. <http://dx.doi.org/10.1007/s10107-004-0559-y>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)