

DOUGLAS–RACHFORD ALGORITHM FOR CONTROL-CONSTRAINED MINIMUM-ENERGY CONTROL PROBLEMS

REGINA S. BURACHIK[✉], BETHANY I. CALDWELL^{*✉} AND
C. YALÇIN KAYA[✉]

Abstract. Splitting and projection-type algorithms have been applied to many optimization problems due to their simplicity and efficiency, but the application of these algorithms to optimal control is less common. In this paper we utilize the Douglas–Rachford (DR) algorithm to solve control-constrained minimum-energy optimal control problems. Instead of the traditional approach where one discretizes the problem and solves it using large-scale finite-dimensional numerical optimization techniques we split the problem in two subproblems and use the DR algorithm to find an optimal point in the intersection of the solution sets of these two subproblems hence giving a solution to the original problem. We derive general expressions for the projections and propose a numerical approach. We obtain analytic closed-form expressions for the projectors of pure, under-, critically- and over-damped harmonic oscillators. We illustrate the working of our approach to solving not only these example problems but also a challenging machine tool manipulator problem. Through numerical case studies, we explore and propose desirable ranges of values of an algorithmic parameter which yield smaller number of iterations.

Mathematics Subject Classification. 49M37; 49N10; 65K10.

Received February 4, 2023. Accepted January 11, 2024.

1. INTRODUCTION

Linear-quadratic (LQ) control problems are an important class of optimal control problems with a quadratic cost (or objective) functional to minimize subject to linear differential equation constraints describing the dynamics – see for theory and applications [1–7]. In this paper we will study applications of projection methods to solving the minimum-energy control of pure, under-, critically- and over-damped harmonic oscillators, as well as a machine tool manipulator, which are all examples of LQ control problems. In fact, in all these applications we impose constraints on the control variable which makes the problems computationally challenging, justifying a novel implementation of projection methods. For the quadratic objective functional, we consider the square norm of the control variable throughout the paper. These problems are what we refer to as *minimum-energy control problems*¹.

Keywords and phrases: Optimal control, harmonic oscillator, Douglas–Rachford algorithm, control constraints, numerical methods.

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

^{*} Corresponding author: bethany.caldwell@mymail.unisa.edu.au

¹It must be stressed that we are not necessarily minimizing the “true” energy of for example a harmonic oscillator per se from a physics point of view. Rather, we are concerned with minimizing the “energy of the control or signal” or the “energy of the force”. Elaboration of this subtle difference in the terminology can also be found in [8], Section 6.17, [9], [10], Section 2.9, Section 5.5 and [11], p. 118.

Projection methods are an emerging field of research in mathematical optimization with successful applications to a wide range of problems, including road design [12], protein reconstruction [13], sphere packing [14], sudoku [15], graph colouring problems [16] and, radiation therapy treatment planning [17]. These methods have chiefly been applied to discrete-time optimal control problems [18], but there has been little or no research into applications to continuous-time optimal control problems, except recently in [19] by Bauschke, Burachik and Kaya. In [19] various projection methods are applied to solve the energy minimizing double integrator problem, where the control variable is constrained, with promising results. The numerical experiments show that projection methods outperform a method employing direct discretization even in solving this relatively simple optimal control problem.

The aforementioned direct discretization approach is to first discretize the problem, typically using a Runge–Kutta method such as the Euler or trapezoidal methods, and then apply finite-dimensional optimization software (for example, AMPL [20] paired with Ipopt [21]) in order to solve the resulting large scale discrete-time optimal control problem. We aim to show the merits of the Douglas–Rachford (DR) algorithm, a popular projection method extended to solving optimization problems. In particular, we aim to solve LQ control problems, which are much more general than the double integrator problem, and compare the DR algorithm with direct discretization.

The approach in this paper exploits the structure of LQ control problems to obtain advantages, just as the approach in [19] does the same with the simple double integrator problem. In our approach we split the constraints of the original LQ problem into two sets: one contains the ODE constraints involving the state variables, and the other contains box constraints on the control variables. These sets are subsets of a Hilbert space, the first one of these subsets constituting a closed affine set (see Cor. 3.5) and the second one a closed and convex set. We define two simpler optimal control subproblems for computing projections, one subject to the affine set and the other to the box. Solutions to these subproblems yield the projectors onto each of the two sets.

The main contributions of this paper are as follows.

- We derive a general expression for the projectors onto the affine and box sets of the minimum-energy control problem. (See Thms. 3.4 and 3.7).
- We obtain closed-form analytical expressions for the projectors of the special problems whose dynamics involve pure as well as under-, critically- and over-damped harmonic oscillators. (See Cors. 3.12–3.18, resp., for projections onto the affine sets of each case, and Corollary 3.10 for projection onto the box).
- The projector expression in Theorem 3.4 necessitates the knowledge of the state transition matrix as well as the Jacobian of the near-miss function of the shooting method. For the case of general minimum-energy control, we present a computational algorithm (namely Alg. 5.1) for constructing the state transition matrix and the Jacobian and thus finding a projector onto the affine set which in turn can be used in general projection algorithms.
- We illustrate the working of Algorithm 5.1 and Theorem 3.4 in the DR algorithm. The DR algorithm is applied to solving not only the above-mentioned example problems but also a challenging machine tool manipulator example problem. These problems should furnish a class of test-bed examples for future studies.
- Selection of an algorithmic parameter plays an important role in the performance of the DR algorithm. Through case studies, by means of the test-bed examples listed above, we explore and propose the ranges of values of this parameter with which the algorithms seem to converge in a smaller number of iterations.

We note that Corollary 3.9, which provides an analytical projector expression in closed-form for the double integrator problem, was originally derived in [19], Proposition 1. Nevertheless, in this paper, we show that this expression can also be obtained using direct substitutions of the state transition matrix and the Jacobian into the general expression in Theorem 3.4.

For all the above-mentioned example problems we perform numerical experiments and compare the performance of the DR algorithm by also using the optimization modelling software AMPL paired with the interior point optimization software Ipopt. In these experiments we observe that not only is the DR algorithm more efficient, *i.e.*, it can find a solution in a much smaller amount of time, than the AMPL–Ipopt suite, but also

that Ipopt sometimes fails in finding a solution at all. We also compare the errors in the control and state variables separately. These cases for different problems are tabulated altogether for an easier appreciation of the conclusions we set out.

The paper is organized as follows. Section 2 contains necessary background and preliminaries on minimum-energy control problems and optimality. In Section 3 we derive the projectors for a general minimum-energy control problem as well as some specific cases. Section 4 presents the DR algorithm that we apply in Section 5. Section 5 provides a numerical approach for obtaining the projector onto the affine set when it is not possible or convenient (due to length) to use an analytical expression. This section also contains numerical experiments comparing the performance of the DR algorithm with a direct discretization approach, as well as an exploration of (in some sense) best values of the parameter of the DR algorithm. Section 6 contains concluding remarks and open problems. In the appendix we provide the detailed proofs of the projectors onto the affine set for the harmonic oscillator problems.

2. MINIMUM-ENERGY CONTROL PROBLEM

In this section we introduce the theoretical framework as well as the optimal control problem we study. We derive the necessary conditions of optimality for the problem via Pontryagin’s maximum principle, which will be instrumental in the derivation of the projectors. We also split the constraints of the problem into two sets which facilitate the projection method we will study.

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 Hilbert space of Lebesgue measurable functions $z : [t_0, t_f] \rightarrow \mathbb{R}^q$, with 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 ℓ_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

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

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 define a general minimum-energy optimal control problem, which is an LQ control problem, as follows.

$$(P) \quad \begin{cases} \min_u & \frac{1}{2} \int_{t_0}^{t_f} \|u(t)\|^2 dt \\ \text{subject to} & \dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0, \quad x(t_f) = x_f, \\ & u(t) \in U \subseteq \mathbb{R}^m, \quad x(t) \in \mathbb{R}^n, \quad \forall t \in [t_0, t_f]. \end{cases}$$

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 set U is a fixed closed subset of $\mathbb{R}^m$. The time varying matrices $A : [t_0, t_f] \rightarrow \mathbb{R}^{n \times n}$ and $B : [t_0, t_f] \rightarrow \mathbb{R}^{n \times m}$ are continuous. The initial and terminal states are

given as x_0 and x_f respectively. Note that, for every $t \in [t_0, t_f]$, we can write

$$B(t)u(t) = \sum_{i=1}^m b_i(t) u_i(t),$$

where $b_i(t) \in \mathbb{R}^n$, $i = 1, \dots, m$, is the i th column of $B(t)$. We note that, when (P) is feasible, it has a unique solution due to the strong convexity of the objective function.

We assume that (i) the dynamical system in (P) is controllable, *i.e.*, by choosing a suitable unconstrained control variable $u(\cdot)$, one can drive any initial state x_0 to any other terminal state x_f , (ii) Problem (P) is feasible, *i.e.*, the constraint set of Problem (P) is nonempty and (iii) Problem (P) is normal, *i.e.*, Pontryagin's maximum principle does not become degenerate and fail to provide information on optimality.

2.1. Optimality conditions

In this section, we use Pontryagin's maximum principle to derive the necessary conditions of optimality for Problem (P).

Various forms of Pontryagin's maximum principle can be found, along with their proofs, in a number of reference books – see, for example, [22], Theorem 1, [23], Chapter 7, [24], Theorem 6.4.1, [25], Theorem 6.37, and [26], Theorem 22.2. We will state Pontryagin's maximum principle using notation and settings from these references. We start by defining the *Hamiltonian function* $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \times [t_0, t_f] \rightarrow \mathbb{R}$ for Problem (P) as

$$H(x(t), u(t), \lambda(t), \lambda_0, t) := \frac{\lambda_0}{2} \|u(t)\|^2 + \lambda(t)^T \left(A(t)x(t) + \sum_{i=1}^m b_i(t) u_i(t) \right),$$

where 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 λ_0 is a real constant. For brevity, we use the following short-hand notation,

$$H[t] := H(x(t), u(t), \lambda(t), \lambda_0, t).$$

The adjoint variable vector is assumed to satisfy the condition (see *e.g.* [23])

$$\dot{\lambda}(t) := -H_x[t] = -A(t)^T \lambda(t) \quad (2.1)$$

for every $t \in [t_0, t_f]$, where $H_x := \partial H / \partial x$. Suppose that the control set U is a box in $\mathbb{R}^m$, *i.e.*, $U = [-a_1, a_1] \times \dots \times [-a_m, a_m]$, $a_i > 0$ for $i = 1, \dots, m$, and 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 Pontryagin's maximum principle asserts that there exist a real number $\lambda_0 \geq 0$ and a continuous adjoint variable vector $\lambda \in \mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^n)$ as defined in Equation (2.1), such that $\lambda(t) \neq \mathbf{0}$ for all $t \in [t_0, t_f]$, and that, for all $t \in [t_0, t_f]$,

$$\begin{aligned} u_i(t) &= \operatorname{argmin}_{|v_i| \leq a_i} H(x(t), u_1(t), \dots, v_i, \dots, u_m(t), \lambda(t), \lambda_0, t) \\ &= \operatorname{argmin}_{|v_i| \leq a_i} \frac{\lambda_0}{2} (u_1(t)^2 + \dots + v_i^2 + \dots + u_m(t)^2) \\ &\quad + \lambda^T(t) (A(t)x(t) + b_1(t) u_1(t) + \dots + b_i(t) v_i + \dots + b_m(t) u_m(t)) \\ &= \operatorname{argmin}_{|v_i| \leq a_i} \frac{\lambda_0}{2} v_i^2 + \lambda^T(t) b_i(t) v_i, \end{aligned} \quad (2.2)$$

for $i = 1, \dots, m$. We ignored all terms that do not depend on v_i to arrive at equation (2.2). If $a_i = \infty$, $i = 1, \dots, m$, *i.e.*, if the control vector is unconstrained, then (2.2) becomes

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

$$\lambda_0 u_i(t) + b_i(t)^T \lambda(t) = 0, \quad (2.3)$$

$i = 1, \dots, m$. We assume that the problem is *normal*, *i.e.*, $\lambda_0 > 0$, so we can take $\lambda_0 = 1$ without loss of generality. Then (2.3) can be solved for $u_i(t)$ as

$$u_i(t) = -b_i(t)^T \lambda(t), \quad (2.4)$$

for $i = 1, \dots, m$; or using the input matrix $B(t)$,

$$u(t) = -B(t)^T \lambda(t). \quad (2.5)$$

With the box constraint on $u(t)$, one gets from (2.2)

$$u_i(t) = \begin{cases} a_i, & \text{if } b_i^T(t) \lambda(t) \leq -a_i, \\ -b_i^T(t) \lambda(t), & \text{if } -a_i \leq b_i^T(t) \lambda(t) \leq a_i, \\ -a_i, & \text{if } b_i^T(t) \lambda(t) \geq a_i, \end{cases} \quad (2.6)$$

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

Recall that the *state transition matrix* $\Phi_A(t, t_0)$ of $\dot{x}(t) = A(t)x(t)$, also referred to as the *resolvent matrix*, is the unique matrix such that $x(t) = \Phi_A(t, t_0)x(t_0)$ – also see [27] for further details and the properties. Then from [28] the solution of the initial value problem $\dot{x}(t) = A(t)x(t) + B(t)u(t)$, $x(t_0) = x_0$, in Problem (P) can simply be written as

$$x(t) = \Phi_A(t, t_0)x_0 + \int_{t_0}^t \Phi_A(t, \tau) B(\tau) u(\tau) d\tau. \quad (2.7)$$

Similarly, equation (2.1) can be solved as $\lambda(t) = \Phi_{(-A^T)}(t, t_0)\lambda_0$, or by using the identity $\Phi_{(-A^T)}(t, t_0) = \Phi_A(t_0, t)^T$ [27], Property 4.5,

$$\lambda(t) = \Phi_A(t_0, t)^T \lambda_0. \quad (2.8)$$

When a_i is small enough so that the control constraint is active it is usually impossible to find an analytical solution for (P), hence the need for numerical methods.

2.2. Constraint splitting

We split the constraints into the two sets given below.

$$\begin{aligned} \mathcal{A} := \{ & u \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m) \mid \exists x \in \mathcal{W}^{1,2}([t_0, t_f]; \mathbb{R}^n) \text{ which solves} \\ & \dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0, \quad x(t_f) = x_f, \quad \forall t \in [t_0, t_f] \}, \end{aligned} \quad (2.9)$$

$$\mathcal{B} := \{ u \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m) \mid -a_i \leq u_i(t) \leq a_i, \quad \forall t \in [t_0, t_f], i = 1, \dots, m \}. \quad (2.10)$$

The set $\mathcal{A}$ is an *affine space* and contains all the feasible control functions from (P) where the control function is unconstrained. The set $\mathcal{B}$ is a *box* which contains all the control functions with components u_i that are constrained by $-a_i$ and a_i (where each a_i is positive). These two sets form the constraint sets for our two subproblems. The reason we split the original problem into two subproblems is because they are much simpler to solve individually so we can derive analytical expressions.

Recall that we assume the dynamical system in (2.9) is *controllable*, i.e., that there exists some control $u(\cdot)$ with which the system can be driven from any x_0 to any other x_f (see [27]), so that $\mathcal{A} \neq \emptyset$. We also assume that $\mathcal{A} \cap \mathcal{B} \neq \emptyset$, namely that Problem (P) is feasible.

3. PROJECTORS

In this section we give the projectors onto the sets $\mathcal{A}$ and $\mathcal{B}$ for a general problem (P) followed by the projectors for some specific problems, namely the double integrator, pure harmonic oscillator and (under-, critically- and over-) damped harmonic oscillator.

3.1. Projectors for general minimum-energy control

Now that we have the constraint sets, we need to define the subproblems. First, recall that the *projection* $P_C(x)$ of a point x onto C is characterized by $P_C(x) \in C$ and, $\forall y \in C$, $\langle y - P_C(x) | x - P_C(x) \rangle \leq 0$ [29], Theorem 3.16. In our context, the projection onto $\mathcal{A}$ from a current iterate u^- is the point u which solves the following problem.

$$(P1) \quad \begin{cases} \min & \frac{1}{2} \int_{t_0}^{t_f} \|u(t) - u^-(t)\|^2 dt = \frac{1}{2} \|u - u^-\|_{\mathcal{L}^2}^2 \\ \text{subject to} & u \in \mathcal{A}. \end{cases}$$

The projection onto $\mathcal{B}$ from a current iterate u^- is the point u which solves the following problem.

$$(P2) \quad \begin{cases} \min & \frac{1}{2} \|u - u^-\|_{\mathcal{L}^2}^2 \\ \text{subject to} & u \in \mathcal{B}. \end{cases}$$

First we provide a technical lemma.

Lemma 3.1. *Given the $n \times n$ matrix $A(t)$, consider the $n^2 \times n^2$ matrix $\tilde{A}(t)$, defined as*

$$\tilde{A}(t) := \begin{bmatrix} A(t) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & A(t) \end{bmatrix},$$

where $\mathbf{0}$ is a zero matrix of appropriate size, and the matrix $A(t)$ appears repeatedly (n times) in diagonal blocks. The state transition matrix of $\tilde{A}(t)$ is the $n^2 \times n^2$ matrix defined as

$$\Phi_{\tilde{A}}(t, t_0) := \begin{bmatrix} \Phi_A(t, t_0) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \Phi_A(t, t_0) \end{bmatrix}, \quad (3.1)$$

where $\Phi_A(t, t_0)$ (the state transition matrix for $A(t)$), appears repeatedly (n times) in diagonal blocks, where all other elements are zero.

Proof. Suppose that $\Phi_A(t, t_0)$ is the state transition matrix of $\dot{y}_i(t) = A(t)y_i(t)$, $i = 1, \dots, n$, where $y_i(t) \in \mathbb{R}^n$. Suppose that $y_i(t_0) = y_{i,0}$, $i = 1, \dots, n$, are the initial conditions. By the definition preceding (2.7), $y_i(t) = \Phi_A(t, t_0)y_{i,0}$ for $i = 1, \dots, n$ is the unique solution. Then with $\tilde{y}(t) := (y_1(t), \dots, y_n(t)) \in \mathbb{R}^{n^2}$, we get $\dot{\tilde{y}}(t) = \tilde{A}(t)\tilde{y}(t)$ and in turn $\Phi_{\tilde{A}}(t, t_0)$ is as required by (3.1) in the lemma. $\square$

Theorem 3.4 further below furnishes an expression for the projector onto $\mathcal{A}$. Even though the proof of this theorem uses a classical shooting technique and broadly follows steps similar to those in [19], Proposition 1, the case considered in Theorem 3.4 is more general. To simplify presentation, we establish next a technical result involving the shooting concept.

Lemma 3.2. Fix $u^- \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m)$ and consider the following initial value problem

$$\begin{aligned} \dot{z}(t, \lambda_0) &= A(t)z(t, \lambda_0) + B(t) [u^-(t) - B(t)^T \Phi_A(t_0, t)^T \lambda_0], \\ z(t_0, \lambda_0) &= x_0, \end{aligned} \quad (3.2)$$

where the dependence of the solution $z(\cdot, \lambda_0)$ on the parameter λ_0 has been made explicit and, with a slight abuse of notation, $\dot{z}(t, \lambda_0) := \partial z(t, \lambda_0) / \partial t$. Then,

- (i) $z(t_f, \cdot)$ is affine (i.e., $z(t_f, \lambda_0)$ is affine in the variable λ_0). In particular, its partial derivative w.r.t. λ_0 is a constant $n \times n$ matrix, which we denote as $\partial z(t_f, 0) / \partial \lambda_0$.
- (ii) Let $z(\cdot, 0)$ be the solution of (3.2) when $\lambda_0 = 0$. Consider the solution $\bar{\lambda}$ of the linear system

$$\frac{\partial z(t_f, 0)}{\partial \lambda_0} \bar{\lambda} = x_f - z(t_f, 0). \quad (3.3)$$

Then $\bar{\lambda}$ verifies

$$\begin{aligned} \dot{z}(t, \bar{\lambda}) &= A(t)z(t, \bar{\lambda}) + B(t) [u^-(t) - B(t)^T \Phi_A(t_0, t)^T \bar{\lambda}], \\ z(t_0, \bar{\lambda}) &= x_0, \\ z(t_f, \bar{\lambda}) &= x_f. \end{aligned} \quad (3.4)$$

Equivalently,

$$u(t) := [u^-(t) - B(t)^T \Phi_A(t_0, t)^T \bar{\lambda}] \in \mathcal{A}. \quad (3.5)$$

Proof. (i) Using (2.1) and (2.8) we can rewrite the dynamics in (3.2) as the system

$$\begin{aligned} \dot{z}(t, \lambda_0) &= A(t)z(t, \lambda_0) + B(t) [u^-(t) - B(t)^T \lambda(t)], \\ \dot{\lambda}(t) &= -A^T(t)\lambda(t), \end{aligned}$$

which, using matrix notation, becomes

$$\begin{bmatrix} \dot{z}(t, \lambda_0) \\ \dot{\lambda}(t) \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)B^T(t) \\ 0_{n \times n} & -A^T(t) \end{bmatrix} \begin{bmatrix} z(t, \lambda_0) \\ \lambda(t) \end{bmatrix} + \begin{bmatrix} B(t) \\ 0_{n \times m} \end{bmatrix} u^-(t). \quad (3.6)$$

Let $g(t, \lambda_0) := [z(t, \lambda_0) \ \lambda(t)]^T$. To show (i) it is enough to prove that $g(t, \cdot)$ is affine in λ_0 . Namely, we claim that

$$g(t, \alpha\lambda_1 + (1 - \alpha)\lambda_2) = \alpha g(t, \lambda_1) + (1 - \alpha)g(t, \lambda_2), \quad (3.7)$$

for all $\alpha \in \mathbb{R}$ and $\lambda_1, \lambda_2 \in \mathbb{R}^n$. Let the first coefficient matrix on the right-hand side of (3.6) be denoted by $C(t)$ and the matrix multiplying $u^-(t)$ be denoted by $D(t)$. Solving (3.6) gives

$$g(t, \lambda_0) = \Phi_C(t, t_0) \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} + \int_{t_0}^t \Phi_C(t, \tau) D(\tau) u^-(\tau) d\tau,$$

where $\Phi_C(t, t_0)$ is the state transition matrix of $\dot{y}(t) = C(t)y(t)$, for all $t \in [t_0, t_f]$. Next we start off with the left-hand side of (3.7), with the aim of getting the right-hand side after direct manipulations.

$$g(t, \alpha\lambda_1 + (1 - \alpha)\lambda_2) = \Phi_C(t, t_0) \begin{bmatrix} x_0 \\ \alpha\lambda_1 + (1 - \alpha)\lambda_2 \end{bmatrix} + \gamma(t),$$

where $\gamma(t) := \int_{t_0}^t \Phi_C(t, \tau) D(\tau) u^-(\tau) d\tau$. Continuing with further manipulations,

$$\begin{aligned} g(t, \alpha\lambda_1 + (1 - \alpha)\lambda_2) &= \Phi_C(t, t_0) \begin{bmatrix} \alpha x_0 + (1 - \alpha)x_0 \\ \alpha\lambda_1 + (1 - \alpha)\lambda_2 \end{bmatrix} + \alpha\gamma(t) + (1 - \alpha)\gamma(t) \\ &= \alpha\Phi_C(t, t_0) \begin{bmatrix} x_0 \\ \lambda_1 \end{bmatrix} + (1 - \alpha)\Phi_C(t, t_0) \begin{bmatrix} x_0 \\ \lambda_2 \end{bmatrix} + \alpha\gamma(t) + (1 - \alpha)\gamma(t) \\ &= \alpha \left(\Phi_C(t, t_0) \begin{bmatrix} x_0 \\ \lambda_1 \end{bmatrix} + \gamma(t) \right) + (1 - \alpha) \left(\Phi_C(t, t_0) \begin{bmatrix} x_0 \\ \lambda_2 \end{bmatrix} + \gamma(t) \right) \\ &= \alpha g(t, \lambda_1) + (1 - \alpha)g(t, \lambda_2), \end{aligned}$$

which verifies (3.7) and thus proves the affineness of $g(t, \cdot)$. Hence $z(t_f, \cdot)$ is affine too. The proof of (i) is complete.

To prove (ii), we use the fact that $z(t_f, \cdot)$ is affine to write

$$z(t_f, \bar{\lambda}) = z(t_f, 0) + \frac{\partial z(t_f, 0)}{\partial \lambda_0} \bar{\lambda},$$

where $\bar{\lambda}$ is as in (3.3) and $\partial z(t_f, 0)/\partial \lambda_0$ is the Jacobian of $z(t_f, \cdot)$ evaluated at $(t_f, 0)$. Equation (3.3) and the above equality yield

$$z(t_f, \bar{\lambda}) = z(t_f, 0) + (x_f - z(t_f, 0)) = x_f.$$

Also note that, by definition, the function $z(\cdot, \bar{\lambda})$ must solve system (3.2) with $\bar{\lambda}$ in place of λ_0 . Altogether, we have shown that $z(\cdot, \bar{\lambda})$ verifies (3.4). The last statement of the lemma now follows from (3.4) and the definition of $\mathcal{A}$. $\square$

The next definition points to the connection between Lemma 3.2 and the shooting method.

Definition 3.3. Define the *near miss function* as

$$\varphi(\lambda_0) := z(t_f, \lambda_0) - x_f, \quad (3.8)$$

where $\lambda_0 \in \mathbf{R}^n$ is arbitrary, and z is as in Lemma 3.2. Namely, $z(t_f, \lambda_0)$ is a solution of system (3.2) evaluated at (t_f, λ_0) . The function φ measures the discrepancy of a solution $z(\cdot, \lambda_0)$ of (3.2) at the end-point $t = t_f$. By Lemma 3.2(i) and (3.8), φ is affine and so its Jacobian $J_\varphi(\lambda_0)$ is a constant matrix such that

$$J_\varphi(\lambda_0) = J_\varphi(0) = \frac{\partial z(t_f, 0)}{\partial \lambda_0}. \quad (3.9)$$

In particular, for every λ_0 we can write

$$\varphi(\lambda_0) = \varphi(0) + J_\varphi(0)\lambda_0.$$

We are now ready to establish our formula for the projection onto $\mathcal{A}$.

Theorem 3.4. *The projection $P_{\mathcal{A}}$ of $u^- \in \mathcal{L}^2([t_0, t_f]; \mathbb{R}^m)$ onto the constraint set $\mathcal{A}$, as the solution of Problem (P1), is given by*

$$P_{\mathcal{A}}(u^-)(t) = u^-(t) - B(t)^T \Phi_A(t_0, t)^T \bar{\lambda}, \quad (3.10)$$

for all $t \in [t_0, t_f]$, where $\bar{\lambda}$ solves

$$\frac{\partial y(t_f)}{\partial \lambda_0} \bar{\lambda} = -(y(t_f) - x_f), \quad (3.11)$$

where

$$y(t_f) := \Phi_A(t_f, t_0) x_0 + \int_{t_0}^{t_f} \Phi_A(t_f, \tau) B(\tau) u^-(\tau) d\tau.$$

Moreover,

$$\frac{\partial y(t_f)}{\partial \lambda_0} = J_\varphi(0), \quad (3.12)$$

where φ and $J_\varphi(0)$ are as in Definition 3.3.

Proof. The Hamiltonian for Problem (P1) is

$$H[t] := \frac{1}{2} \|u(t) - u^-(t)\|^2 + \lambda^T(t)(A(t)x(t) + B(t)u(t)).$$

From Pontryagin's maximum principle, $H_u[t] = 0$ and so

$$u(t) = u^-(t) - B(t)^T \lambda(t), \quad (3.13)$$

for all $t \in [t_0, t_f]$. To fully solve this equation, we need to find $\lambda(\cdot)$ such that the function u on the left hand-side of (3.13) belongs to $\mathcal{A}$. Equation (2.8) gives $\lambda(\cdot)$ as a function of an initial condition λ_0 . The aim is therefore to determine an initial condition such that the corresponding $\lambda(\cdot)$ produces a function u in $\mathcal{A}$. To avoid confusion,

we call $\bar{\lambda}$ this desired initial condition. We now use the last statement of Lemma 3.2, which states that $\bar{\lambda}$ as in (3.3) ensures that u as in (3.5) is in $\mathcal{A}$. Altogether, this choice of $\bar{\lambda}$ verifies

$$u(t) = u^-(t) - B(t)^T \lambda(t) = u^-(t) - B(t)^T \Phi_A(t_0, t)^T \bar{\lambda} \in \mathcal{A}, \quad (3.14)$$

where we used (3.5) in the inclusion and in the second equality we used (2.8) with $\bar{\lambda}$ in place of λ_0 . Hence, u is the desired projection onto $\mathcal{A}$ for this choice of $\bar{\lambda}$.

To establish the rest of the theorem, call $y(\cdot) := z(\cdot, 0)$, where $z(\cdot, 0)$ is a solution of the IVP (3.2) for $\lambda_0 := 0$. Therefore, we have

$$y(t_f) = \Phi_A(t_f, t_0) x_0 + \int_{t_0}^{t_f} \Phi_A(t_f, \tau) B(\tau) u^-(\tau) d\tau. \quad (3.15)$$

Since $y(\cdot) = z(\cdot, 0)$ we have that

$$\frac{\partial z(t_f, 0)}{\partial \lambda_0} = \frac{\partial y(t_f)}{\partial \lambda_0},$$

so condition (3.3) in the lemma becomes (3.11). The last statement of the theorem follows from the above equality and (3.9). $\square$

Corollary 3.5. *Let X be a Hilbert space and let $C \subset X$ be a nonempty convex set. Assume that for every $u \in X$ there exists the projection $P_C(u)$ of u onto C . Then the set C must be closed.*

Proof. Denote by $\text{cl } C$ the closure of C and take any $z \in \text{cl } C$. By assumption on C , the projection $P_C(z)$ of z onto C exists. The definitions imply that

$$\|z - P_C(z)\| = d(z, C) = d(z, \text{cl } C) = 0.$$

Hence, $z = P_C(z) \in C$. So $\text{cl } C \subset C$ and therefore C is closed. $\square$

Remark 3.6. One can find the elements of $\partial y(t_f)/\partial \lambda_0 = J_\varphi(0)$ by solving the variational equations

$$\frac{\partial z(t, \lambda_0)}{\partial t} = A(t)z(t, \lambda_0) + B(t) [u^-(t) - B(t)^T \Phi_A(t_0, t)^T \lambda_0], \quad z(t_0, \lambda_0) = x_0. \quad (3.16)$$

with respect to λ_0 , i.e., by solving the following equations in $(\partial z/\partial \lambda_{0,i})(t, \lambda_0) \in \mathbb{R}^n$, for $i = 1, \dots, n$.

$$\frac{\partial}{\partial t} \left(\frac{\partial z}{\partial \lambda_{0,i}} \right) (t, \lambda_0) = A(t) \frac{\partial z}{\partial \lambda_{0,i}} (t, \lambda_0) - B(t) B(t)^T \Phi_A(t_0, t)^T e_i$$

where $e_i \in \mathbb{R}^n$ are the canonical basis vectors, i.e., with 1 in the i th coordinate and zero elsewhere. Let $\tilde{y}(t), \dot{\tilde{y}}(t) \in \mathbb{R}^{n^2}$ where

$$\tilde{y} := \begin{bmatrix} \partial z / \partial \lambda_{0,1} \\ \vdots \\ \partial z / \partial \lambda_{0,n} \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \text{and} \quad \dot{\tilde{y}} := \begin{bmatrix} \frac{\partial}{\partial t} (\partial z / \partial \lambda_{0,1}) \\ \vdots \\ \frac{\partial}{\partial t} (\partial z / \partial \lambda_{0,n}) \end{bmatrix} = \begin{bmatrix} \dot{y}_1 \\ \vdots \\ \dot{y}_n \end{bmatrix},$$

then $\dot{\tilde{y}} = \tilde{A}(t)\tilde{y} + \tilde{B}(t)$, $\tilde{y}(t_0) = 0$ where

$$\tilde{A}(t) = \begin{bmatrix} A(t) & \mathbf{0} \\ & \ddots \\ \mathbf{0} & A(t) \end{bmatrix} \in \mathbb{R}^{n^2 \times n^2} \quad \text{and} \quad \tilde{B}(t) = -B(t)B(t)^T \Phi_A(t_0, t)^T \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} \in \mathbb{R}^{n^2}.$$

Using Lemma 3.1 along with knowledge of differential equations

$$\tilde{y}(t) = \int_{t_0}^t \Phi_{\tilde{A}}(t, \tau) \tilde{B}(\tau) d\tau, \quad (3.17)$$

where $\Phi_{\tilde{A}}(t, t_0)$ is the transition matrix of $\dot{\tilde{y}} = \tilde{A}(t)\tilde{y}$. So evaluating the above integral and substituting $t = t_f$ gives the components of $\partial y(t_f)/\partial \lambda_0$. $\square$

Theorem 3.7 (Projection onto $\mathcal{B}$). *The projection $P_{\mathcal{B}}$ of $u^- \in \mathcal{L}^2([t_0, t_f]; \mathbb{R})$ onto the constraint set $\mathcal{B}$, as the solution of Problem (P2), is given by*

$$[P_{\mathcal{B}}(u^-)(t)]_i = \begin{cases} a_i, & \text{if } u_i^-(t) \geq a_i, \\ u_i^-(t), & \text{if } -a_i \leq u_i^-(t) \leq a_i, \\ -a_i, & \text{if } u_i^-(t) \leq -a_i, \end{cases} \quad (3.18)$$

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

Proof. Simply use separability of Problem (P2) in u_i , $i = 1, \dots, m$. $\square$

3.2. Projectors for special cases

In this subsection we consider problems with two state variables ($n = 2$) and one control variable ($m = 1$). In particular we consider problems involving the double integrator and the pure and damped harmonic oscillators, for which the general system and control matrices in set $\mathcal{A}$ in (2.9) become

$$A(t) = A = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix} \quad \text{and} \quad B(t) = b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (3.19)$$

where ω_0 is the natural frequency and ζ is the damping ratio. Note that $\zeta = 0$ is the case of pure (undamped) harmonic oscillator, and $0 < \zeta < 1$ under-damped, $\zeta = 1$ critically-damped, and $\zeta > 1$ over-damped harmonic oscillator. The general forms of the constraint sets can be found in (2.9)–(2.10) but for this specialization we define

$$\begin{aligned} \mathcal{A}_{\omega_0, \zeta} &:= \{u \in \mathcal{L}^2([0, t_f]; \mathbb{R}) \mid \exists x \in \mathcal{W}^{1,2}([0, t_f]; \mathbb{R}^2) \text{ which solves} \\ &\quad \dot{x}_1(t) = x_2(t), \quad x_1(0) = s_0, \quad x_1(t_f) = s_f, \\ &\quad \dot{x}_2(t) = -\omega_0^2 x_1(t) - 2\zeta\omega_0 x_2(t) + u(t), \quad x_2(0) = v_0, \quad x_2(t_f) = v_f, \\ &\quad \forall t \in [0, t_f]\}, \end{aligned} \quad (3.20)$$

$$\mathcal{B} := \{u \in \mathcal{L}^2([0, t_f]; \mathbb{R}) \mid -a \leq u(t) \leq a, \quad \forall t \in [0, t_f]\}. \quad (3.21)$$

To maintain the flow of this paper we move the proofs from this subsection to Appendix A as these are rather lengthy and the proof techniques follow a similar pattern. In the lemmas we complete some of the technical steps by deriving expressions for the state transition matrices and Jacobians required in Theorem 3.4. Then the corollaries follow by direct substitution into the expression (3.10) in Theorem 3.4. Since we find analytical expressions in each of the lemmas we express the inverse of the Jacobian and use it directly in the expression in Theorem 3.4.

In the case where the inverse of the Jacobian is analytical and not lengthy we express λ_0 as

$$\lambda_0 = -[J_\varphi(0)]^{-1}(x(t_f) - x_f)$$

to have a more closed form expression for the projector:

$$P_{\mathcal{A}}(u^-)(t) = u^-(t) + B(t)^T \Phi_A(t_0, t)^T [J_\varphi(0)]^{-1} \left(\Phi_A(t_f, t_0) x_0 + \int_{t_0}^{t_f} \Phi_A(t_f, \tau) B(\tau) u^-(\tau) d\tau - x_f \right). \quad (3.22)$$

In the cases of the double integrator as well as the under-, critically- and over-damped harmonic oscillators the inverse of the Jacobian is simple enough, so we will use (3.22).

3.2.1. Double integrator

The dynamics of the double integrator are given by $\ddot{y}(t) = f(t)$, where $f(t)$ stands for forcing, which typically models the motion of a point mass (or analogously, an electric circuit or a fluid system with capacitance) – see pertaining references in [19], where $y(t)$ is the position and $\dot{y}(t)$ the velocity at time t . With $x_1 := y$ and $x_2 := \dot{y}$, one gets the state equations $\dot{x}_1 = x_2$ and $\dot{x}_2 = u$; in other words, $\omega_0 = 0$ and $\zeta = 0$, resulting in the constraint set $\mathcal{A}_{0,0}$. We note from (3.19) that

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The minimum-energy problem that we consider corresponds, for example to the practical problem of engineering where one would like to minimize the average magnitude of the force, or the problem of designing cubic (variational) curves.

In what follows we present the projections onto $\mathcal{A}_{0,0}$ and $\mathcal{B}$ in the Corollaries 3.9 and 3.10 below. These two results and their proofs can be found in [19].

Recall the definition of the state transition matrix $\Phi_A(t, t_0)$ via (2.7) and the definition of the Jacobian $J_\varphi(0)$ in (3.9). The following lemma evaluates $\Phi_A(t, 0)$ and $[J_\varphi(0)]^{-1}$ for the double integrator, which are utilized in the proof of Corollary 3.9.

Lemma 3.8 (Computation of Φ_A and J_φ for $\omega_0 = 0$, $\zeta = 0$). *One has that*

$$\Phi_A(t, 0) = e^{At} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}, \quad [J_\varphi(0)]^{-1} = \begin{bmatrix} -12 & 6 \\ -6 & 2 \end{bmatrix}. \quad (3.23)$$

Proof. See [proof](#) in Appendix A. □

The following result is a corollary of Theorem 3.4. As mentioned already, this result along with its proof can be found in [19], but we present here a new proof which directly substitutes the expressions in Lemma 3.8 into Theorem 3.4.

Corollary 3.9 (Projection onto $\mathcal{A}_{0,0}$ [19]). *The projection $P_{\mathcal{A}_{0,0}}$ of $u^- \in \mathcal{L}^2([0, 1]; \mathbb{R})$ onto the constraint set $\mathcal{A}_{0,0}$, as the solution of Problem (P1) with $\omega_0 = 0$ and $\zeta = 0$, is given by*

$$P_{\mathcal{A}_{0,0}}(u^-)(t) = u^-(t) + c_1 t + c_2, \quad (3.24)$$

for all $t \in [0, 1]$, where

$$c_1 := 12 \left(s_0 + v_0 - s_f + \int_0^1 (1 - \tau) u^-(\tau) d\tau \right) - 6 \left(v_0 - v_f + \int_0^1 u^-(\tau) d\tau \right), \quad (3.25)$$

$$c_2 := -6 \left(s_0 + v_0 - s_f + \int_0^1 (1 - \tau) u^-(\tau) d\tau \right) + 2 \left(v_0 - v_f + \int_0^1 u^-(\tau) d\tau \right). \quad (3.26)$$

Proof. See [proof](#) in Appendix A. □

The following result is a direct consequence of Theorem 3.7 for all cases of the harmonic oscillator.

Corollary 3.10 (Projection onto $\mathcal{B}$ [19]). *The projection $P_{\mathcal{B}}$ of $u^- \in \mathcal{L}^2([0, t_f]; \mathbb{R})$ onto the constraint set $\mathcal{B}$, as the solution of Problem (P2), is given by*

$$P_{\mathcal{B}}(u^-)(t) = \begin{cases} a, & \text{if } u^-(t) \geq a, \\ u^-(t), & \text{if } -a \leq u^-(t) \leq a, \\ -a, & \text{if } u^-(t) \leq -a, \end{cases} \quad (3.27)$$

for all $t \in [0, t_f]$.

3.2.2. Pure harmonic oscillator

When a spring is added to the point mass, or an inductor to the electric circuit with a capacitor, one gets the *pure* (or undamped) harmonic oscillator, as without forcing, once excited the state variables will exhibit sustained oscillations (or sinusoids) at frequency ω_0 . We extend Corollary 3.9 to the general case of projecting onto $\mathcal{A}_{\omega_0,0}$. In this case, from (3.19) one has

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix}. \quad (3.28)$$

The following lemma provides major ingredients for the projector in Theorem 3.4.

Lemma 3.11 (Computation of Φ_A and J_φ for $\omega_0 > 0$, $\zeta = 0$). *One has that*

$$\Phi_A(t, 0) = e^{At} = \begin{bmatrix} \cos(\omega_0 t) & \frac{\sin(\omega_0 t)}{\omega_0} \\ -\omega_0 \sin(\omega_0 t) & \cos(\omega_0 t) \end{bmatrix}, \quad [J_\varphi(0)]^{-1} = \begin{bmatrix} -\omega_0^2/\pi & 0 \\ 0 & -1/\pi \end{bmatrix}. \quad (3.29)$$

Proof. See [proof](#) in Appendix A. □

The following corollary is a direct consequence of Theorem 3.4.

Corollary 3.12 (Projection onto $\mathcal{A}_{\omega_0,0}$). *The projection $P_{\mathcal{A}_{\omega_0,0}}$ of $u^- \in \mathcal{L}^2([0, 2\pi]; \mathbb{R})$ onto the constraint set $\mathcal{A}_{\omega_0,0}$, as the solution of Problem (P1), $\zeta = 0$, is given by*

$$P_{\mathcal{A}_{\omega_0,0}}(u^-)(t) = u^-(t) + c_1 \sin(\omega_0 t) - c_2 \cos(\omega_0 t), \quad (3.30)$$

where

$$c_1 := \frac{\omega_0}{\pi} \left(s_0 - s_f - \frac{1}{\omega_0} \int_0^{2\pi} \sin(\omega_0 \tau) u^-(\tau) d\tau \right),$$

$$c_2 := \frac{1}{\pi} \left(v_0 - v_f + \int_0^{2\pi} \cos(\omega_0 \tau) u^-(\tau) d\tau \right).$$

Proof. See [proof](#) in Appendix A. □

3.2.3. Damped harmonic oscillator

If a damper (which is an element that dissipates energy) is added to the mass-spring system (or analogously a resistor added to a capacitor–inductor electrical circuit) one gets what is referred to as a *damped harmonic oscillator*. There are three cases to consider for a damped system, namely the *critically-* ($\zeta = 1$), *over-* ($\zeta > 1$) and *under-damped* ($0 < \zeta < 1$) cases. We provide the projectors for each case.

Corollary 3.14 below presents the projector onto the set $\mathcal{A}_{\omega_0,1}$ for the critically-damped case. From (3.19), the system matrix A for this case is

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\omega_0 \end{bmatrix}.$$

Lemma 3.13 (Computation of Φ_A and J_φ for $\omega_0 > 0$, $\zeta = 1$). *One has that*

$$\Phi_A(t, 0) = e^{At} = e^{-\omega_0 t} \begin{bmatrix} \omega_0 t + 1 & t \\ -t\omega_0^2 & -\omega_0 t + 1 \end{bmatrix} \quad (3.31)$$

and

$$[J_\varphi(0)]^{-1} = \frac{1}{y_{11}(2\pi)y_{22}(2\pi) - y_{12}(2\pi)y_{21}(2\pi)} \begin{bmatrix} y_{22}(2\pi) & -y_{21}(2\pi) \\ -y_{12}(2\pi) & y_{11}(2\pi) \end{bmatrix} \quad (3.32)$$

where $y_{ij}(2\pi)$, $j = 1, 2$, are the components of the vectors $y_i(2\pi)$, $i = 1, 2$, given below.

$$y(2\pi) = \begin{bmatrix} y_1(2\pi) \\ y_2(2\pi) \end{bmatrix} = \begin{bmatrix} \frac{e^{-2\pi\omega_0}(2\pi\omega_0 - e^{4\pi\omega_0} + 2\pi\omega_0 e^{4\pi\omega_0} + 1)}{4\omega_0^3} \\ \frac{\pi \sinh(2\pi\omega_0)}{\omega_0} \\ \dots\dots\dots \\ -\frac{\pi \sinh(2\pi\omega_0)}{\omega_0} \\ -\frac{e^{-2\pi\omega_0}(2\pi\omega_0 + e^{4\pi\omega_0} + 2\pi\omega_0 e^{4\pi\omega_0} - 1)}{4\omega_0} \end{bmatrix}. \quad (3.33)$$

Proof. See [proof](#) in Appendix A. $\square$

Corollary 3.14 (Projection onto $\mathcal{A}_{\omega_0,1}$). *The projection $P_{\mathcal{A}_{\omega_0,\zeta}}$ of $u^- \in \mathcal{L}^2([0, 2\pi]; \mathbb{R})$ onto the constraint set $\mathcal{A}_{\omega_0,1}$, as the solution of Problem (P1) with $\zeta = 1$, is given by*

$$P_{\mathcal{A}_{\omega_0,1}}(u^-)(t) = u^-(t) + \frac{e^{\omega_0(t-2\pi)}}{y_{11}y_{22} - y_{12}y_{21}} \left(-((y_{22} + y_{12}\omega_0)t + y_{12}) \left(x_1(2\pi) - \frac{s_f}{e^{-2\pi\omega_0}} \right) \right. \\ \left. + ((y_{21} + y_{11}\omega_0)t + y_{11}) \left(x_2(2\pi) - \frac{v_f}{e^{-2\pi\omega_0}} \right) \right), \quad (3.34)$$

where y_{ij} are the components of $y(2\pi)$ given in (3.33).

Proof. See [proof](#) in Appendix A. $\square$

In Corollary 3.16 we consider the derivation of the projection onto the set $\mathcal{A}_{\omega_0,\zeta}$ from (3.20) where $\zeta > 1$.

Lemma 3.15 (Computation of Φ_A and J_φ for $\omega_0 > 0, \zeta > 1$). *One has that*

$$\Phi_A(t, 0) = e^{At} = \frac{e^{-\alpha t}}{\beta} \begin{bmatrix} \omega_0 \sinh(\beta t + \eta) & \sinh(\beta t) \\ -\omega_0^2 \sinh(\beta t) & \omega_0 \sinh(-\beta t + \eta) \end{bmatrix} \quad (3.35)$$

with $\alpha = \omega_0\zeta$, $\beta = \omega_0\sqrt{\zeta^2 - 1}$, $\eta = \frac{1}{2} \ln \left| \frac{\beta + \alpha}{\beta - \alpha} \right|$. Then we express the inverse of the Jacobian as in (3.32) where $y_{ij}(2\pi)$, $j = 1, 2$, are the components of the vectors $y_i(2\pi)$, $i = 1, 2$, given below.

$$y(2\pi) = \begin{bmatrix} y_1(2\pi) \\ y_2(2\pi) \end{bmatrix} = \begin{bmatrix} \frac{e^{-2\pi\alpha}}{4\omega_0^2} \left(\frac{(1 - e^{4\pi\alpha}) \cosh(2\pi\beta)}{\alpha} + \frac{(1 + e^{4\pi\alpha}) \sinh(2\pi\beta)}{\beta} \right) \\ \frac{\sinh(2\pi\alpha) \sinh(2\pi\beta)}{2\alpha\beta} \\ \text{-----} \\ -\frac{\sinh(2\pi\alpha) \sinh(2\pi\beta)}{2\alpha\beta} \\ \frac{e^{-2\pi\alpha}}{4} \left(\frac{(1 - e^{4\pi\alpha}) \cosh(2\pi\beta)}{\alpha} + \frac{(1 + e^{4\pi\alpha}) \sinh(2\pi\beta)}{\beta} \right) \end{bmatrix}. \quad (3.36)$$

Proof. See [proof](#) in Appendix A. $\square$

Corollary 3.16 (Projection onto $\mathcal{A}_{\omega_0,\zeta}$ with $\zeta > 1$). *The projection $P_{\mathcal{A}_{\omega_0,\zeta}}$ of $u^- \in \mathcal{L}^2([0, 2\pi]; \mathbb{R})$ onto the constraint set $\mathcal{A}_{\omega_0,\zeta}$, as the solution of Problem (P1) where $\zeta > 1$, is given by*

$$P_{\mathcal{A}_{\omega_0,\zeta}}(u^-)(t) = u^-(t) + \frac{e^{\alpha(t-2\pi)}}{\beta^2(y_{11}(2\pi)y_{22}(2\pi) - y_{12}(2\pi)y_{21}(2\pi))} \left(- (y_{22}(2\pi) \sinh(\beta t) + y_{12}(2\pi) \right. \\ \left. \times \omega_0 \sinh(\beta t + \eta)) \left(x_1(2\pi) - \frac{\beta s_f}{e^{-2\pi\alpha}} \right) \right. \\ \left. + (y_{21}(2\pi) \sinh(\beta t) + y_{11}(2\pi) \omega_0 \sinh(\beta t + \eta)) \left(x_2(2\pi) - \frac{\beta v_f}{e^{-2\pi\alpha}} \right) \right) \quad (3.37)$$

where $\alpha = \omega_0\zeta$, $\beta = \omega_0\sqrt{\zeta^2 - 1}$ and y_{ij} are the components of $y(2\pi)$ given in (3.36).

Proof. See [proof](#) in Appendix A. $\square$

In Corollary 3.18 we consider the final case for the damped harmonic oscillator, which is the projection onto the set $\mathcal{A}_{\omega_0, \zeta}$ from (3.20) where $0 < \zeta < 1$.

Lemma 3.17 (Computation of Φ_A and J_φ for $\omega_0 > 0, 0 < \zeta < 1$). *One has that*

$$\Phi_A(t, 0) = e^{At} = \frac{e^{-\alpha t}}{\tilde{\beta}} \begin{bmatrix} \omega_0 \cos(\tilde{\beta}t + \gamma) & \sin(\tilde{\beta}t) \\ -\omega_0^2 \sin(\tilde{\beta}t) & \omega_0 \cos(\tilde{\beta}t - \gamma) \end{bmatrix}, \quad (3.38)$$

where $\alpha = \omega_0 \zeta$, $\tilde{\beta} = \omega_0 \sqrt{1 - \zeta^2}$, $\gamma = \tan^{-1}(-\frac{\alpha}{\tilde{\beta}})$. Then we express the inverse of the Jacobian as in (3.32) where $y_{ij}(2\pi)$, $j = 1, 2$, are the components of the vectors $y_i(2\pi)$, $i = 1, 2$, given below.

$$y(2\pi) = \begin{bmatrix} y_1(2\pi) \\ y_2(2\pi) \end{bmatrix} = \begin{bmatrix} \frac{e^{-2\pi\alpha}}{4\omega_0^2} \left(\frac{\cos(2\pi\tilde{\beta})(1 - e^{4\pi\alpha})}{\alpha} + \frac{\sin(2\pi\tilde{\beta})(1 + e^{4\pi\alpha})}{\tilde{\beta}} \right) \\ \frac{\sinh(2\pi\alpha) \sin(2\pi\tilde{\beta})}{2\alpha\tilde{\beta}} \\ \dots\dots\dots \frac{\sinh(2\pi\alpha) \sin(2\pi\tilde{\beta})}{2\alpha\tilde{\beta}} \\ \frac{e^{-2\pi\alpha}}{4} \left(\frac{\cos(2\pi\tilde{\beta})(1 - e^{4\pi\alpha})}{\alpha} - \frac{\sin(2\pi\tilde{\beta})(1 + e^{4\pi\alpha})}{\tilde{\beta}} \right) \end{bmatrix}. \quad (3.39)$$

Proof. See [proof](#) in Appendix A. □

Corollary 3.18 (Projection onto $\mathcal{A}_{\omega_0, \zeta}$ with $0 < \zeta < 1$). *The projection $P_{\mathcal{A}_{\omega_0, \zeta}}$ of $u^- \in \mathcal{L}^2([0, 2\pi]; \mathbb{R})$ onto the constraint set $\mathcal{A}_{\omega_0, \zeta}$, as the solution of Problem (P1) where $0 < \zeta < 1$, is given by*

$$\begin{aligned} P_{\mathcal{A}_{\omega_0, \zeta}} u^- &= u^-(t) + \frac{e^{\alpha(t-2\pi)}}{\tilde{\beta}^2(y_{11}(2\pi)y_{22}(2\pi) - y_{12}(2\pi)y_{21}(2\pi))} \\ &\quad \left(- \left(y_{22}(2\pi) \sin(\tilde{\beta}t) + y_{12}(2\pi) \omega_0 \cos(\tilde{\beta}t + \gamma) \right) \left(x_1(2\pi) - \frac{\tilde{\beta}s_f}{e^{-2\pi\alpha}} \right) \right. \\ &\quad \left. + \left(y_{21}(2\pi) \sin(\tilde{\beta}t) + y_{11}(2\pi) \omega_0 \cos(\tilde{\beta}t + \gamma) \right) \left(x_2(2\pi) - \frac{\tilde{\beta}v_f}{e^{-2\pi\alpha}} \right) \right), \end{aligned}$$

where $\alpha = \omega_0 \zeta$, $\tilde{\beta} = \omega_0 \sqrt{1 - \zeta^2}$ and y_{ij} are the components of $y(2\pi)$ given in (3.39).

Proof. See [proof](#) in Appendix A. □

Remark 3.19. Note that Corollary 3.12 cannot be recovered from Corollary 3.18 by simply taking $\zeta \rightarrow 0$. Similarly, Corollary 3.14 cannot be recovered from Corollary 3.16 or Corollary 3.18 by taking $\zeta \rightarrow 1$.

3.3. Machine tool manipulator

A machine tool manipulator is an automatic machine that simulates human hand operations. The dynamics of this machine can be formulated as a LQ control problem as in [4] – also see [2]. For this problem the system

and control matrices in (2.9) become

$$A(t) = A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -4.441 \times 10^7/450 & 0 & 0 & -8500/450 & 0 & 0 & -1/450 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/750 \\ 0 & 0 & -8.2 \times 10^6/40 & 0 & 0 & -1800/40 & 0.25/40 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/0.0025 \end{bmatrix},$$

$$B(t) = b = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/0.0025 \end{bmatrix}^T,$$

$x_0 = (0, 0, 0, 0, 0, 0, 0)$, $x_f = (0, 2.7 \times 10^{-3}, 0, 0, 0.1, 0, 0)$, $t_0 = 0$, $t_f = 0.0522$. Unlike the special cases in the previous subsection we will not provide analytical projectors for this problem. Because this problem has 7 state variables computing Φ_A and $J_\phi(0)$ analytically is not a simple task. Instead we will introduce and implement the numerical procedure in Section 5.

4. DOUGLAS–RACHFORD ALGORITHM

The Douglas–Rachford algorithm, in our context, is a projection algorithm, which we recall here by closely following the framework in [19]. We consider a real Hilbert space denoted by X , with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$. We will consider the sets $\mathcal{A}$ and $\mathcal{B}$ to align with the previous results but note that the only assumptions required are that $\mathcal{A}$ is a closed affine subspace of X and $\mathcal{B}$ is a nonempty closed convex subset of X .

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 [29], Definition 12.23:

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

for any $u \in L^2([t_0, t_f]; \mathbb{R}^m)$. We also recall that the *indicator function* ι_C of C is given by

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

Note that $\text{Prox}_{\iota_C} = P_C$. Given $\beta > 0$, we specialize the DR algorithm (see [30], [31] and [32]) to the case of minimizing the sum of the two functions $f(x) := \iota_{\mathcal{B}}(x) + \frac{\beta}{2} \|x - z\|^2$ and $g := \iota_{\mathcal{A}}$. For this case, the DR operator becomes

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

Given f, g we know that the respective proximal mappings are $\text{Prox}_f(x) = P_{\mathcal{B}}(\frac{1}{1+\beta}x + \frac{\beta}{1+\beta}z)$ and $\text{Prox}_g = P_{\mathcal{A}}$ (see [29], Prop. 24.8(i)). Set $\lambda := \frac{1}{1+\beta} \in]0, 1[$. It follows that the DR operator becomes

$$Tx = x - P_{\mathcal{B}}(\lambda x + (1 - \lambda)z) + P_{\mathcal{A}}(2P_{\mathcal{B}}(\lambda x + (1 - \lambda)z) - x). \quad (4.1)$$

Now fix $x_0 \in X$ and let $z := 0$. Given $x_n \in X$, $n \geq 0$, update

$$b_n := P_{\mathcal{B}}(\lambda x_n), \quad x_{n+1} := Tx_n = x_n - b_n + P_{\mathcal{A}}(2b_n - x_n). \quad (4.2)$$

Using [29], Corollary 28.3(v)(a) we have that $(b_n)_{n \in \mathbb{N}}$ converges strongly to the unique solution of Problem (P). Observe that strong convergence is due to the strong convexity of the function f , and is for the sequence $(b_n)_{n \in \mathbb{N}}$ and not necessarily $(x_n)_{n \in \mathbb{N}}$. By definition of Problem (P), the unique solution of (P) is the element of minimum norm in $\mathcal{A} \cap \mathcal{B}$. Namely, we have that the limit of the sequence $(b_n)_{n \in \mathbb{N}}$ is $x = P_{\mathcal{A} \cap \mathcal{B}}(0)$. We point out that, in general, only weak convergence is guaranteed for this method (see [33], Thm. 1 or [34], Thm. 4.4).

Note that (4.2) simplifies to

$$x_{n+1} := x_n - P_{\mathcal{B}}(\lambda x_n) + P_{\mathcal{A}}(2P_{\mathcal{B}}(\lambda x_n) - x_n) \quad \text{provided that } z = 0. \quad (4.3)$$

See Algorithm 4.1 below for a step-by-step description of the numerical implementation.

Algorithm 4.1. (DR)

- Step 1** (*Initialization*) Choose a parameter $\lambda \in]0, 1[$ and the initial iterate u^0 arbitrarily. Choose a small parameter $\varepsilon > 0$, and set $k = 0$.
Step 2 (*Projection onto $\mathcal{B}$*) Set $u^- = \lambda u^k$. Compute $\tilde{u} = P_{\mathcal{B}}(u^-)$ by using (3.18).
Step 3 (*Projection onto $\mathcal{A}$*) Set $u^- := 2\tilde{u} - u^k$. Compute $\hat{u} = P_{\mathcal{A}}(u^-)$ by using (3.10) or Algorithm 5.1.
Step 4 (*Update*) Set $u^{k+1} := u^k + \hat{u} - \tilde{u}$.
Step 5 (*Stopping criterion*) If $\|u^{k+1} - u^k\|_{L^\infty} \leq \varepsilon$, then return $\tilde{u}$ and stop. Otherwise, set $k := k + 1$ and go to Step 2.

Remark 4.2. Robustness of the DR algorithm is supported by the fact that many inexact versions of it are shown to converge as well, see [35, 36]. In [37] we see a study of the complexity of an inexact version of the algorithm. This justifies the use of discrete approximations of the function iterates in our implementation. $\square$

Remark 4.3. The convergence properties of the DR algorithm for the case when $\mathcal{A} \cap \mathcal{B} = \emptyset$, *i.e.*, when Problem (P) is infeasible, have been studied recently by Bauschke and Moursi [38]. A study of this interesting case for optimal control might be an promising direction to pursue in the future. $\square$

5. NUMERICAL APPROACH

In Subsection 3.2 we have a selection of problems where we have derived analytical expressions for the projection onto $\mathcal{A}$. In practice however the state transition matrix may be too difficult (if not impossible) to find analytically, in which case one needs to employ a numerical technique, as will be outlined further below. Following the presentation of this algorithm we give numerical experiments used to choose the optimal values of the parameter λ for the DR algorithm and compare the performance of DR with the AMPL-Ipopt suite.

5.1. Background and algorithm for projector onto $\mathcal{A}$

From Equation (3.13) we can see that in order to define the projection we must find λ . In Theorem 3.4 we assumed that $\Phi_A(t_0, t)$ is available. We can see from (2.8) that the knowledge of $\Phi_A(t_0, t)$ is necessary to find λ . In the case where we cannot find the state transition matrix directly to substitute into (2.8), we must solve

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)B^T(t) \\ 0_{n \times n} & -A^T(t) \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} + \begin{bmatrix} B(t) \\ 0_{n \times m} \end{bmatrix} u^-(t), \quad (5.1)$$

for all $t \in [t_0, t_f]$, with the initial conditions (ICs) $x(t_0) = x_0$ and $x(t_f) = x_f$, to find λ . Throughout the steps of Algorithm 5.1, we will solve the linear system (5.1) with different ICs. The ICs that we will consider are

$$(i) \begin{bmatrix} x(t_0) \\ \lambda(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}, \quad (ii) \begin{bmatrix} x(t_0) \\ \lambda(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ e_i \end{bmatrix}, \quad (iii) \begin{bmatrix} x(t_0) \\ \lambda(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix}. \quad (5.2)$$

As in the proof of Theorem 3.4 we define $z(t, \lambda_0) := x(t)$. Recall in this case that $\dot{x}(t) = dx(t)/dt$ can be written as $\partial z(t, \lambda_0)/\partial t$. We also recall that the near-miss function $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ as defined in (3.8) is affine by Lemma 3.2(i) and Definition 3.3. Then the Taylor series expansion of φ about zero is simply

$$\varphi(\lambda_0) = \varphi(0) + J_\varphi(0)\lambda_0.$$

Substituting (3.8), one gets

$$z(t_f, \lambda_0) = z(t_f, 0) + J_\varphi(0)\lambda_0,$$

and, rearranging,

$$J_\varphi(0)\lambda_0 = z(t_f, \lambda_0) - z(t_f, 0).$$

Suppose $\lambda_0 = e_i$. Then

$$J_\varphi(0)e_i = z(t_f, e_i) - z(t_f, 0), \quad (5.3)$$

which is the i th column of $J_\varphi(0)$. Therefore, by finding $z(t_f, 0)$ and $z(t_f, e_i)$ for every $i = 1, \dots, n$ we can build the Jacobian $J_\varphi(0)$. Consequently, a procedure for constructing $J_\varphi(0)$ can be prescribed as follows.

1. Solve (5.1) with ICs (ii) in (5.2) to get $z(t_f, e_i)$.
2. Solve (5.1) with ICs (i) in (5.2) to get $z(t_f, 0)$.
3. Compute the i th column of $J_\varphi(0)$ using (5.3), for $i = 1, \dots, n$, and obtain $J_\varphi(0)$.

As in the proof of Lemma 3.2, we can now solve the linear system

$$J_\varphi(0)\lambda_0 = -\varphi(0) = -(z(t_f, 0) - x_f), \quad (5.4)$$

for λ_0 , since in the procedure for finding $J_\varphi(0)$ we have computed all the other components of this equation. Then once we have λ_0 we can solve (5.1) with ICs (iii) in (5.2) to obtain λ .

The algorithm below describes the steps for computing the projection of a current iterate u^- onto the constraint set $\mathcal{A}$. In solving (5.1) with each of the ICs in (5.2) we implement MATLAB's numerical ODE solver `ode45` or a direct implementation of some Runge–Kutta method such as the Euler method.

Algorithm 5.1. (Numerical Computation of the Projector onto $\mathcal{A}$)

Step 0 (*Initialization*) The following 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.

Step 1 (*Near-miss function*) Solve (5.1) with ICs in (5.2)(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 (5.1) with ICs in (5.2)(ii), to get $z(t_f, e_i)$.

Set $\beta_i(t) := z(t_f, e_i) - z(t_f, 0)$ and $J_\varphi(0) := [\beta_1(t) \mid \dots \mid \beta_n(t)]$.

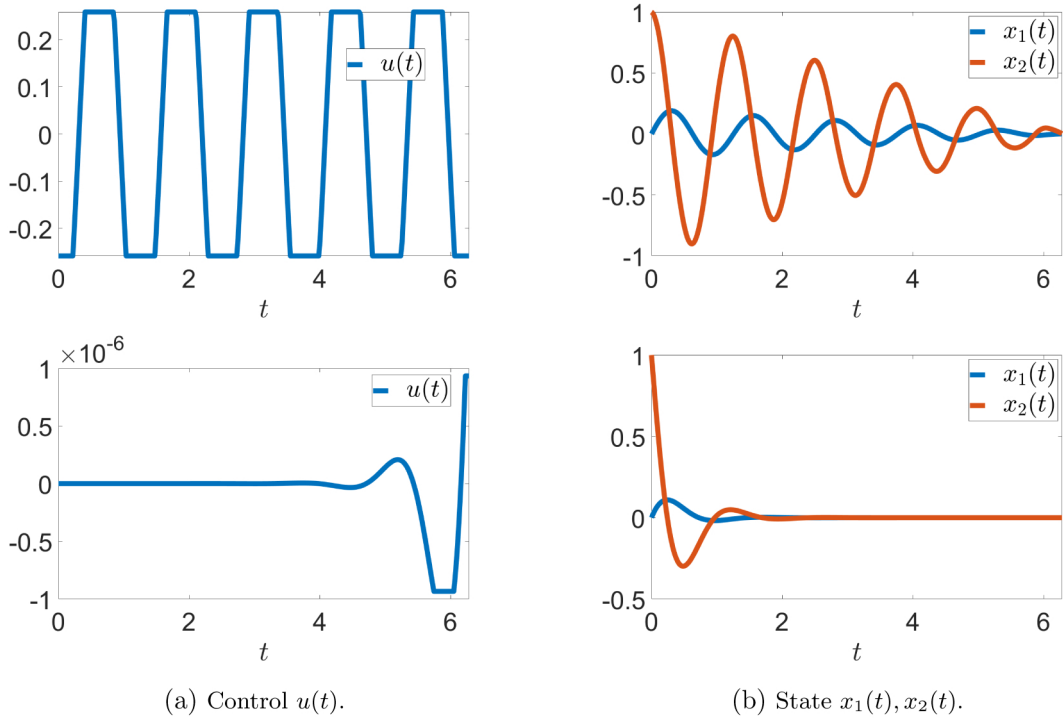


FIGURE 1. Top plots for $\omega_0 = 5$, $\zeta = 0$ where $|u(t)| \leq 0.259$. Bottom plots for $\omega_0 = 5$, $\zeta = 0.5$ where $|u(t)| \leq 9.34 \times 10^{-7}$.

Step 3 (*Missing IC*) Solve $J_\varphi(0) \lambda_0 := -\varphi(0)$ for λ_0 .

Step 4 (*Projector onto $\mathcal{A}$*) Solve (5.1) with ICs in (5.2)(iii) to find $\lambda(t)$.

Set $P_{\mathcal{A}}(u^-)(t) := u^-(t) - B^T(t)\lambda(t)$.

5.2. Experiments

For computations in this section we use MATLAB release R2021b for implementing the DR algorithm and error analysis. We also use AMPL–Ipopt computational suite [20, 21] (with Ipopt version 3.12.13) for comparison with the DR algorithm since the suit is commonly used for solving similar optimal control problems.

In Figure 1 we have the pure and under-damped oscillator solution plots for the constrained control where $\omega_0 = 5$. The boundary conditions are $x_2(0) = 1$ and $x_1(0) = x_1(2\pi) = x_2(2\pi) = 0$. The bound on u for the under-damped case is much smaller than the value used in the pure case to ensure that the control constraint is active. In Figure 2, we display the control variable solution plot for the machine tool manipulator with $|u(t)| \leq 2000$.

5.2.1. Parameter plots

In Section 4 the DR algorithm requires a choice of $\lambda \in]0, 1[$. In Figure 3 we experiment with different parameter choices for the pure harmonic oscillator, under-damped harmonic oscillator and machine tool manipulator.

In Figure 3 we see, for different bounds on u , plots with the number of iterations taken by DR for different values of the parameter λ . In each of these plots five values for the bound on u were taken, the smallest bound being close to the value that will lead to a problem with no solution and the largest resulting in an unconstrained u . These plots give information on the “best” value of λ to choose that produce the smallest number of iterations. This is advantageous because a reduction in the number of iterations will result in a reduction in run time.

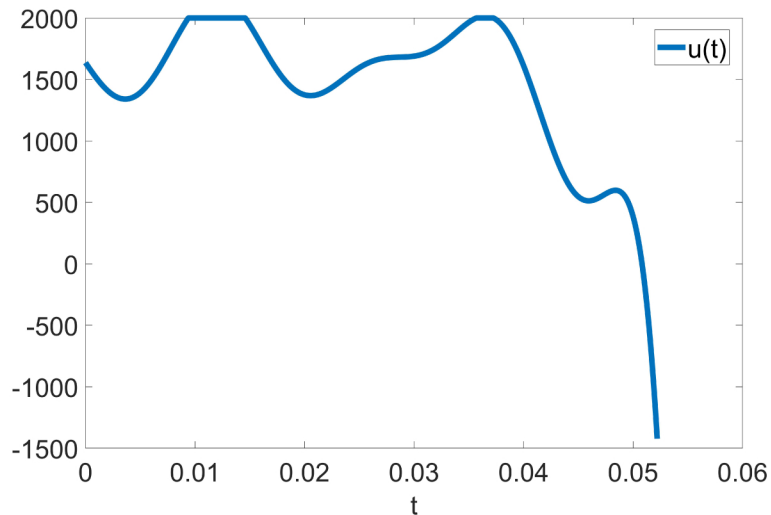


FIGURE 2. Control solution plot for the machine tool manipulator where $|u(t)| \leq 2000$.

Figure 3 contains the experiments for the harmonic oscillator with various values of the pair (ω_0, ζ) , as well as the machine tool manipulator. We see from Figure 3a that when $\zeta = 0$ varying ω_0 does not seem to have an impact on the best choice for λ , in fact the curves numerically found over the values $1 \leq \omega_0 \leq 100$ seem to be identical. For example when $|u(t)| \leq 0.259$ the “best” value for λ looks to be approximately 0.7 for any $1 \leq \omega_0 \leq 100$. We observe downward spikes at some parameter values that result in a large decrease in iterations. Although these spikes achieve a much smaller number of iterations we would not necessarily select these values in practice because a slight shift from the parameter values results in a large increase in the number of iterations.

In Figures 3b–3d we have the parameter graphs for the under-damped harmonic oscillator with $\zeta = 0.5$ and $\omega_0 = 1, 3$ and 5. This time there are differences in the curves for different values of ω_0 . For this damped problem the larger the value of ω_0 , the closer the optimal u gets to zero; see Figure 1. Hence the values of a are chosen to be much smaller for $\omega_0 = 3, 5$ so the control constraint remains active though again the largest value of a , given by the enveloping lowermost (green) curve, is the case of unconstrained u .

The behaviour of the cases with $\omega_0 = 1$ and 3 are very similar with both having some spikes present, the optimal λ value for the unconstrained u case being almost 1 and bumps present in the curves. Though for $\omega_0 = 5$ we see some odd behaviour for the three largest values of a . When using the lowermost (green) curve, *i.e.*, when u is unconstrained, the number of iterations greatly increases at $\lambda \approx 0.98$. We also observe for the yellow and purple curves that the number of iterations seems to level off before rapidly increasing which is something we don’t observe in the other cases. These anomalies could potentially be numerical artifacts but further investigation is needed to draw a conclusion.

For the machine tool manipulator in Figure 3e we again see a lot of similarities to the figures from the other problems. Although for the machine tool manipulator we see many small ripples in the curves and for the blue curve there are almost no parameter choices where the methods converge in less than 300 iterations. The blue curves represent the problem where a is so small that there are almost no solutions to the problem. So it is fair to say that when the problem is almost infeasible it is impossible to get a solution in reasonable time.

In general we observe some similarities across all the problems in Figure 3. For all the problems the “best” parameter choices are $\lambda \geq 0.5$. As we approach the critical value of a where the problems have no solution the “best” choice of λ approaches 1.

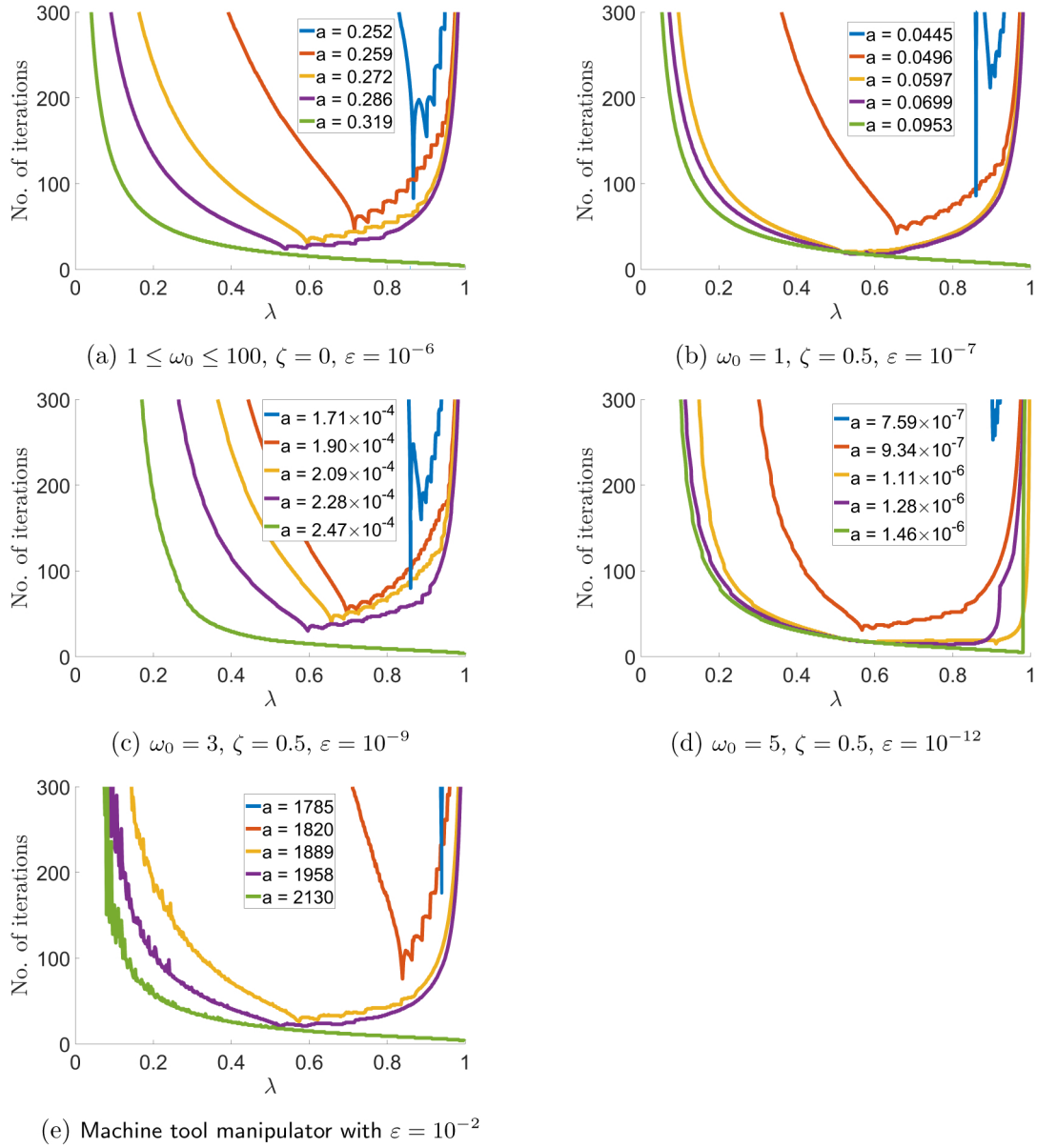


FIGURE 3. Parameter curves for the harmonic oscillator with various values of (ω_0, ζ) and for the machine tool manipulator with tolerance values.

5.2.2. Error and CPU time comparisons

The iterates of the DR algorithm (Alg. 4.1) are functions and in each iteration function addition and scalar multiplication operations need to be performed. Obviously we can perform these operations numerically only on approximations of functions. For approximations we consider discretization of the function iterates in that the iterates are represented by N discrete values over a regular partition of their domains.

TABLE 1. Tolerance, bounds on control variable and parameter choices for numerical experiments. MTM stands for machine tool manipulator.

(ω_0, ζ)	ε	a	λ
(1, 0)	10^{-6}	0.259	0.75
(5, 0)	10^{-6}	0.259	0.75
(1, 0.5)	10^{-7}	4.96×10^{-2}	0.65
(5, 0.5)	10^{-12}	9.34×10^{-7}	0.6
MTM	10^{-2}	2000	0.55

The AMPL–Ipopt suite is, on the other hand, already a numerical scheme for finite-dimensional optimization problems and as such it is applied to the *direct (Euler) discretization* (see e.g. [39]) of Problem (P), with the same N so that the discrete solutions obtained by the DR algorithm and the AMPL–Ipopt suite can be compared.

In Step 5 of Algorithm 4.1 we use as stopping criterion the difference between two consecutive iterates in function space. In the implementation of Algorithm 4.1, discretized iterates are used so in turn Step 5 uses finite dimensional norm to evaluate this stopping criterion.

We rather compute *a posteriori* the absolute true errors in the solution depending on N . Namely if u_N denotes the approximate (discretized) solution of control and u_N^* the discretized exact/true solution, then the error is the maximum of the absolute difference, in other words, $\|u_N - u_N^*\|_{\ell_\infty}$. The fact that the stopping criterion in Step 5 is effective is shown by the fact that the actual absolute error tends to zero as N grows.

In Table 2 we display these errors as well as the CPU times with the number discretization points $N = 10^3, 10^4$ and 10^5 for all the previously mentioned problems with the specifications given in Table 1. Since we cannot find analytical solutions for these problems the “true” solution u^* we are comparing to in this error analysis was computed using the DR algorithm with $N = 10^7$ and tolerance 10^{-12} .

We are mostly interested in the errors in the control variable since that is the variable being optimized. The states are computed as an auxiliary process using the optimal control found and Euler’s method. In Table 2 we see that for the DR algorithm, in general, an increase in the discretization points used by some order results in a decrease in the error by the same order. This is useful to know because if a particular error is required the number of discretization points needed to reach that accuracy can easily be determined.

The only case where this seems to differ is in the machine tool manipulator example. For example with $N = 10^3$ the error with the DR algorithm is 9.3×10^1 so following the observed pattern we would expect that when $N = 10^5$ the error should be around 9.3×10^{-1} but instead we have 5.5×10^{-1} . A possible explanation may be that the expected error is not far enough from the “true” solution. Or since the machine tool manipulator is the only example to use the numerical implementation rather than analytical expressions the errors are different to what we expected. Whatever the reason we see that the machine tool manipulator is still achieving at least one order of improvement in error for the control variable.

In Table 2 we also observe that for the states we have the same relationship between the orders of the number of discretization points and error for the DR algorithm. In the states the errors for the DR algorithm are much closer to that of Ipopt, with Ipopt resulting in better error in some of the cases. These differences in performance of the DR algorithm in the state variables could be because of the extra errors introduced from Euler’s method in the computation of the states. If we were to use a more accurate method to compute the states it is possible we would have less discrepancies, though by implementing a more complicated method the run time would increase.

In the fifth and sixth columns of Table 2 we have the CPU times. In these calculations the CPU times recorded are averages from 1,000 runs on a PC with an i5-10500T 2.30 GHz processor and 8GB RAM. On average the DR algorithm is more than 10 times faster, with some cases of the DR algorithm being as much as 200 or more times faster than AMPL–Ipopt. In general for the DR algorithm an increase in the number of discretization points results in a proportional increase in run time. We can use this observation to estimate the CPU time for any number of discretization points.

TABLE 2. Errors in control and states and CPU times for the DR algorithm and AMPL–Ipopt, with specifications from Table 1. For Ipopt we set the tolerance `tol` to 10^{-6} . A dash means a method was unsuccessful in getting a solution. MTM stands for machine tool manipulator.

N	(ω_0, ζ)	L^∞ error in control		L^∞ error in states		CPU time [sec]	
		DR	Ipopt	DR	Ipopt	DR	Ipopt
10^3	(1, 0)	4.0×10^{-3}	4.2×10^{-2}	1.5×10^{-2}	1.3×10^{-2}	4.2×10^{-3}	2.4×10^{-1}
	(5, 0)	1.8×10^{-2}	—	3.5×10^{-1}	—	4.5×10^{-3}	—
	(1, 0.5)	2.1×10^{-3}	6.8×10^{-3}	4.3×10^{-3}	5.1×10^{-3}	5.5×10^{-3}	2.3×10^{-1}
	(5, 0.5)	1.2×10^{-7}	1.5×10^{-6}	1.5×10^{-2}	1.5×10^{-2}	4.3×10^{-3}	1.9×10^{-1}
	MTM	9.3×10^1	4.6×10^1	2.8×10^1	2.9×10^1	1.0×10^{-1}	1.2×10^0
10^4	(1, 0)	4.0×10^{-4}	1.4×10^{-2}	1.5×10^{-3}	3.1×10^{-3}	4.9×10^{-2}	2.1×10^0
	(5, 0)	1.8×10^{-3}	1.2×10^{-1}	2.6×10^{-2}	7.5×10^{-3}	5.4×10^{-2}	1.3×10^{-1}
	(1, 0.5)	2.1×10^{-4}	1.0×10^{-2}	4.3×10^{-4}	9.3×10^{-3}	4.7×10^{-2}	1.9×10^0
	(5, 0.5)	1.2×10^{-8}	8.2×10^{-7}	1.5×10^{-3}	1.5×10^{-3}	4.1×10^{-2}	1.5×10^0
	MTM	9.2×10^0	—	2.9×10^0	—	1.1×10^0	—
10^5	(1, 0)	4.0×10^{-5}	2.5×10^{-1}	1.5×10^{-4}	5.8×10^{-2}	4.2×10^{-1}	8.5×10^1
	(5, 0)	1.7×10^{-4}	7.6×10^{-2}	2.5×10^{-3}	3.1×10^{-3}	4.8×10^{-1}	1.5×10^0
	(1, 0.5)	2.1×10^{-5}	1.8×10^{-2}	4.3×10^{-5}	1.8×10^{-2}	4.1×10^{-1}	1.8×10^1
	(5, 0.5)	1.2×10^{-9}	8.4×10^{-7}	1.5×10^{-4}	1.5×10^{-4}	3.7×10^{-1}	1.5×10^1
	MTM	5.5×10^{-1}	—	2.6×10^{-1}	—	9.5×10^1	—

6. CONCLUSION AND OPEN PROBLEMS

We have derived general expressions for the projectors respectively onto the affine set and the box of the minimum-energy control problem. We provided closed-form expressions for the pure, critically-, over- and under-damped harmonic oscillators. For problems where we do not have the necessary information to use the general expression for the projector onto the affine set, we proposed a numerical scheme to compute the projection. In our numerical experiments we have applied this numerical scheme to solve a machine tool manipulator problem. We carried out numerical experiments with all the previously mentioned problems, the closed-form examples and the machine tool manipulator, comparing the errors and CPU times. These numerical experiments compared the performance of the DR algorithm with the AMPL–Ipopt suite.

For the DR algorithm we collected some numerical results regarding the use of different values for the parameter λ and its effect on the number of iterations required for the method to converge. In this parameter analysis we observed that as the bounds on the control variable are tightened the choice of parameter becomes more difficult. We also noticed that when the problem is almost infeasible, *i.e.*, the bounds on the control variable are so tight that almost no solutions exist, the parameter value approaches 1.

Regarding our other numerical experiments we observe that an increase in the order of discretization points produces a resulting decrease in the order of the errors, both for the control and state variables. We also see that an increase in the order of discretization points results in an increase in the order of the CPU time. These observations are useful to estimate the run time and errors for any number of discretization points and were not seen in the results from Ipopt. In general we see smaller errors and faster CPU times when using the DR algorithm. Overall, using the DR algorithm with the general expressions and numerical approach we proposed is more advantageous than using Ipopt for the class of problems we consider.

In the future it would be useful to extend this research to more general problems. One such extension is the case when the ODE constraints are nonlinear. If the ODE constraints are nonlinear then we have a nonconvex problem. The DR algorithm has already been shown to have success with finite-dimensional nonconvex problems so it would be interesting to apply this method to nonconvex minimum-energy control problems. An extension to LQ problems where the objective function is given by

$$\frac{1}{2} \int_{t_0}^{t_f} \left(x(t)^T Q(t) x(t) + u(t)^T R(t) u(t) \right) dt,$$

where Q, R are positive semi-definite and positive definite matrix functions of dimensions $n \times n$ and $m \times m$, respectively, should also be investigated. The fact that the state variables appear in the objective makes this problem particularly interesting and challenging to study.

Another possibility is to look into the cause of the intriguing numerical results in Figure 3d where $\omega_0 = 5$, $\zeta = 0.5$, using the DR algorithm. It is currently unknown whether the behaviour in this case is a result of a theoretical fact or it is just a numerical artifact.

In an earlier preprint version of this paper on arXiv [40], various other projection methods are tested as well as the DR algorithm on the same optimal control problems as in this paper. These additional methods are namely the method of alternating projections (MAP) [41, 42] and the Dykstra [43] and Aragón Artacho–Campoy (AAC) [44] algorithms. MAP consists of sequential, or alternating, projections onto each of the sets $\mathcal{A}$ and $\mathcal{B}$, and Dykstra is some modification of MAP. However, MAP can only find a point in $\mathcal{A} \cap \mathcal{B}$. Unlike DR or Dykstra, MAP cannot handle an objective function that is not an indicator. Performance comparisons with another projection method, namely the AAC algorithm, a special case of which is the DR algorithm, can also be found in [40]. We chose to focus on a single projection algorithm here, namely the DR algorithm, for which convergence theory involving convex optimization problems in Hilbert spaces have been well studied and cited widely in the literature. In the future it would be interesting to implement other projection algorithms such as the Peaceman–Rachford algorithm or the projected gradient method.

APPENDIX A.

In this appendix we provide the proofs of the lemmas and corollaries from Section 3.2.

Proof of Lemma 3.8

With the A given for the double integrator, it is straightforward to compute the expressions in (3.23). To find $J_\varphi(0)$ we need to solve Equation (3.17) with

$$\tilde{A} = \begin{bmatrix} 0 & 1 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 \\ \hline 0 & 0 & \vdots & 0 & 1 \\ 0 & 0 & \vdots & 0 & 0 \end{bmatrix} \quad \tilde{b} = \begin{bmatrix} 0 \\ t \\ \hline 0 \\ -1 \end{bmatrix}.$$

Using Equation (3.17)

$$y(1) = \begin{bmatrix} y_1(1) \\ \hline y_2(1) \end{bmatrix} = \int_0^1 \begin{bmatrix} 1 & 1-\tau & \vdots & 0 & 0 \\ 0 & 1 & \vdots & 0 & 0 \\ \hline 0 & 0 & \vdots & 1 & 1-\tau \\ 0 & 0 & \vdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \tau \\ \hline 0 \\ -1 \end{bmatrix} d\tau = \begin{bmatrix} 1/6 \\ 1/2 \\ \hline -1/2 \\ -1 \end{bmatrix}.$$

This gives us

$$J_\varphi(0) = \begin{bmatrix} y_1(1) & y_2(1) \end{bmatrix} = \begin{bmatrix} 1/6 & -1/2 \\ 1/2 & -1 \end{bmatrix},$$

when inverted we arrive at the expression in (3.23). □

Proof of Corollary 3.9

Recall the results for the state transition matrices and the Jacobian in Lemma 3.8. By direct substitution of these quantities into (3.22), we get

$$\begin{aligned} P_{\mathcal{A}_{0,0}}(u^-)(t) &= u^-(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ -t & 1 \end{bmatrix} \begin{bmatrix} -12 & 6 \\ -6 & 2 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_0 \\ v_0 \end{bmatrix} \right. \\ &\quad \left. + \int_0^1 \begin{bmatrix} 1 & 1-\tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ u^-(\tau) \end{bmatrix} d\tau - \begin{bmatrix} s_f \\ v_f \end{bmatrix} \right), \\ &= u^-(t) + \begin{bmatrix} 12t-6 & -6t+2 \end{bmatrix} \left(\begin{bmatrix} s_0+v_0-s_f \\ v_0-v_f \end{bmatrix} + \int_0^1 \begin{bmatrix} (1-\tau)u^-(\tau) \\ u^-(\tau) \end{bmatrix} d\tau \right), \\ &= u^-(t) + \left(12 \left(s_0+v_0-s_f + \int_0^1 (1-\tau)u^-(\tau) d\tau \right) \right. \\ &\quad \left. - 6 \left(v_0-v_f + \int_0^1 u^-(\tau) d\tau \right) \right) t \\ &\quad - 6 \left(s_0+v_0-s_f + \int_0^1 (1-\tau)u^-(\tau) d\tau \right) + 2 \left(v_0-v_f + \int_0^1 u^-(\tau) d\tau \right), \end{aligned}$$

as required. □

Proof of Lemma 3.11

Given A for the harmonic oscillator, it is straightforward to compute the state transition matrices in (3.29). To find $J_\varphi(0)$ we need to solve Equation (3.17) where

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_0^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_0^2 & 0 \end{bmatrix} \quad \tilde{b} = \begin{bmatrix} 0 \\ \sin(\omega_0 t)/\omega_0 \\ 0 \\ -\cos(\omega_0 t) \end{bmatrix}.$$

Using Equation (3.17),

$$y(2\pi) = \begin{bmatrix} y_1(2\pi) \\ y_2(2\pi) \end{bmatrix} = \int_0^{2\pi} \begin{bmatrix} e^{A(2\pi-\tau)} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & e^{A(2\pi-\tau)} \end{bmatrix} \begin{bmatrix} 0 \\ \sin(\omega_0 \tau)/\omega_0 \\ 0 \\ -\cos(\omega_0 \tau) \end{bmatrix} d\tau = \begin{bmatrix} -\pi/\omega_0^2 \\ 0 \\ 0 \\ -\pi \end{bmatrix}.$$

As in Lemma 3.8 this gives us

$$J_\varphi(0) = \begin{bmatrix} y_1(2\pi) & \vdots & y_2(2\pi) \end{bmatrix} = \begin{bmatrix} -\pi/\omega_0^2 & \vdots & 0 \\ 0 & \vdots & -\pi \end{bmatrix}$$

and the expression for the inverse Jacobian in (3.29) follows. $\square$

Proof of Corollary 3.12

Recall the results from Lemma 3.11. By direct substitution of the state transition matrix and Jacobian into (3.22), we get

$$\begin{aligned} P_{\mathcal{A}_{0,0}}(u^-)(t) &= u^-(t) + \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\omega_0 t) & \omega_0 \sin(\omega_0 t) \\ -\frac{\sin(\omega_0 t)}{\omega_0} & \cos(\omega_0 t) \end{bmatrix} \begin{bmatrix} -\frac{\omega_0^2}{\pi} & 0 \\ 0 & -\frac{1}{\pi} \end{bmatrix} \\ &\quad \left(\begin{bmatrix} \cos(2\pi\omega_0) & \frac{\sin(2\pi\omega_0)}{\omega_0} \\ -\omega_0 \sin(2\pi\omega_0) & \cos(2\pi\omega_0) \end{bmatrix} \begin{bmatrix} s_0 \\ v_0 \end{bmatrix} \right. \\ &\quad \left. + \int_0^{2\pi} \begin{bmatrix} \cos(\omega_0(2\pi - \tau)) & \frac{\sin(\omega_0(2\pi - \tau))}{\omega_0} \\ -\omega_0 \sin(\omega_0(2\pi - \tau)) & \cos(\omega_0(2\pi - \tau)) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} u^-(\tau) d\tau - \begin{bmatrix} s_f \\ v_f \end{bmatrix} \right), \\ &= u^-(t) + \begin{bmatrix} \frac{\omega_0 \sin(\omega_0 t)}{\pi} \\ -\frac{\cos(\omega_0 t)}{\pi} \end{bmatrix}^T \left(\begin{bmatrix} s_0 - s_f \\ v_0 - v_f \end{bmatrix} \int_0^{2\pi} \begin{bmatrix} -u^-(\tau) \frac{\sin(\omega_0 \tau)}{\omega_0} \\ u^-(\tau) \cos(\omega_0 \tau) \end{bmatrix} d\tau \right), \\ &= u^-(t) + \frac{\omega_0}{\pi} \left(s_0 - s_f - \frac{1}{\omega_0} \int_0^{2\pi} \sin(\omega_0 \tau) u^-(\tau) d\tau \right) \sin(\omega_0 t) \\ &\quad - \frac{1}{\pi} \left(v_0 - v_f + \int_0^{2\pi} \cos(\omega_0 \tau) u^-(\tau) d\tau \right) \cos(\omega_0 t), \end{aligned}$$

as stated. $\square$

Proof of Lemma 3.13

Given A for the damped harmonic oscillator with $\zeta = 1$, it is straightforward to compute the state transition matrix in (3.31). To find $J_\varphi(0)$ we need to solve Equation (3.17) with

$$\tilde{A} = \begin{bmatrix} 0 & 1 & \vdots & 0 & 0 \\ -\omega_0^2 & -2\omega_0 & \vdots & 0 & 0 \\ \hline 0 & 0 & \vdots & 0 & 1 \\ 0 & 0 & \vdots & -\omega_0^2 & -2\omega_0 \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} 0 \\ te^{\omega_0 t} \\ \hline 0 \\ -e^{\omega_0 t}(\omega_0 t + 1) \end{bmatrix}.$$

Using Equation (3.17)

$$y(2\pi) = \begin{bmatrix} y_1(2\pi) \\ y_2(2\pi) \end{bmatrix} = \int_0^{2\pi} \begin{bmatrix} e^{A(2\pi-\tau)} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & e^{A(2\pi-\tau)} \end{bmatrix} \begin{bmatrix} 0 \\ \tau e^{\omega_0 \tau} \\ 0 \\ -e^{\omega_0 \tau}(\omega_0 \tau + 1) \end{bmatrix} d\tau$$

yields, after integration, the required expression in (3.33). So in the same way as in the proofs of Lemmas 3.8–3.11

$$J_\varphi(0) = \begin{bmatrix} y_1(2\pi) & y_2(2\pi) \end{bmatrix},$$

and once inverted the expression in (3.32) follows. $\square$

Proof of Corollary 3.14

We begin by finding $x(2\pi)$.

$$\begin{aligned} \begin{bmatrix} x_1(2\pi) \\ x_2(2\pi) \end{bmatrix} &= e^{-2\pi\omega_0} \begin{bmatrix} 2\pi\omega_0 + 1 & 2\pi \\ -2\pi\omega_0^2 & -2\pi\omega_0 + 1 \end{bmatrix} \begin{bmatrix} s_0 \\ v_0 \end{bmatrix} \\ &\quad + \int_0^{2\pi} e^{-(2\pi-\tau)\omega_0} \begin{bmatrix} (2\pi-\tau)\omega_0 + 1 & (2\pi-\tau) \\ -(2\pi-\tau)\omega_0^2 & -(2\pi-\tau)\omega_0 + 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} u^-(\tau) d\tau, \\ &= e^{-2\pi\omega_0} \begin{bmatrix} s_0(2\pi\omega_0 + 1) + 2\pi v_0 + \int_0^{2\pi} u^-(\tau)(2\pi-\tau)e^{\omega_0 \tau} d\tau \\ -2\pi s_0\omega_0^2 - v_0(2\pi\omega_0 - 1) - \int_0^{2\pi} u^-(\tau)((2\pi-\tau)\omega_0 - 1)e^{\omega_0 \tau} d\tau \end{bmatrix}. \end{aligned}$$

Recall the results from Lemma 3.13. By direct substitution of the state transition matrix and Jacobian into (3.22), we get

$$\begin{aligned} P_{A_{\omega_0,1}}(u^-)(t) &= u^-(t) + \begin{bmatrix} 0 & 1 \end{bmatrix} e^{\omega_0 t} \begin{bmatrix} -\omega_0 t + 1 & t\omega_0^2 \\ -t & \omega_0 t + 1 \end{bmatrix} \frac{1}{y_{11}(2\pi)y_{22}(2\pi) - y_{12}(2\pi)y_{21}(2\pi)} \\ &\quad \times \begin{bmatrix} y_{22}(2\pi) & -y_{21}(2\pi) \\ -y_{12}(2\pi) & y_{11}(2\pi) \end{bmatrix} \left(\begin{bmatrix} x_1(2\pi) \\ x_2(2\pi) \end{bmatrix} - \begin{bmatrix} s_f \\ v_f \end{bmatrix} \right). \end{aligned}$$

After carrying out some of the matrix multiplications on the right-hand side one gets

$$P_{A_{\omega_0,1}}(u^-)(t) = u^-(t) + \frac{e^{\omega_0(t-2\pi)}}{y_{11}y_{22} - y_{12}y_{21}} \begin{bmatrix} -y_{22}t - y_{12}(\omega_0 t + 1) \\ -y_{21}t + y_{11}(\omega_0 t + 1) \end{bmatrix}^T \begin{bmatrix} x_1(2\pi) - s_f \\ x_2(2\pi) - v_f \end{bmatrix},$$

where we have omitted the arguments for y to save space. Expansions and further manipulations yield the required expression. $\square$

Proof of Lemma 3.15

Given A for the damped harmonic oscillator with $\zeta > 1$, it is straightforward to compute the state transition

matrices in (3.35). To find $J_\varphi(0)$ we need to solve Equation (3.17) where

$$\tilde{A} = \begin{bmatrix} 0 & 1 & \vdots & 0 & 0 \\ -\omega_0^2 & -2\omega_0\zeta & \vdots & 0 & 0 \\ \hline 0 & 0 & \vdots & 0 & 1 \\ 0 & 0 & \vdots & -\omega_0^2 & -2\omega_0\zeta \end{bmatrix} \quad \tilde{b} = \frac{e^{\alpha t}}{\beta} \begin{bmatrix} 0 \\ \sinh(\beta t) \\ \hline 0 \\ -\omega_0 \sinh(\beta t + \eta) \end{bmatrix}.$$

Using Equation (3.17)

$$y(2\pi) = \frac{e^{\alpha\tau}}{\beta} \int_0^{2\pi} \begin{bmatrix} e^{A(2\pi-\tau)} & \vdots & \mathbf{0}_{2 \times 2} \\ \hline \mathbf{0}_{2 \times 2} & \vdots & e^{A(2\pi-\tau)} \end{bmatrix} \begin{bmatrix} 0 \\ e^{\alpha\tau} \sinh(\beta\tau) \\ \hline 0 \\ -\omega_0 \sinh(\beta\tau + \eta) \end{bmatrix} d\tau$$

which, after integration, yields (3.36). As in Lemmas 3.8–3.13

$$J_\varphi(0) = \begin{bmatrix} y_1(2\pi) & \vdots & y_2(2\pi) \end{bmatrix},$$

inversion of which results in the expression in (3.32). □

Proof of Corollary 3.16

We begin by finding $x(t_f)$.

$$\begin{aligned} x(2\pi) &= \frac{e^{-2\pi\alpha}}{\beta} \begin{bmatrix} \omega_0 \sinh(2\pi\beta + \eta) & \sinh(2\pi\beta) \\ -\omega_0^2 \sinh(2\pi\beta) & \omega_0 \sinh(-2\pi\beta + \eta) \end{bmatrix} \begin{bmatrix} s_0 \\ v_0 \end{bmatrix} \\ &+ \int_0^{2\pi} \frac{e^{-\alpha(2\pi-\tau)}}{\beta} \begin{bmatrix} u^-(\tau) \sinh(\beta(2\pi - \tau)) \\ u^-(\tau) \omega_0 \sinh(-\beta(2\pi - \tau) + \eta) \end{bmatrix} d\tau, \\ &= \frac{e^{-\alpha(2\pi-\tau)}}{\beta} \begin{bmatrix} s_0 \omega_0 \sinh(2\pi\beta + \eta) + v_0 \sinh(2\pi\beta) + C \\ -s_0 \omega_0 \sinh(2\pi\beta) + v_0 \omega_0 \sinh(-2\pi\beta + \eta) + D \end{bmatrix}, \end{aligned}$$

where

$$\begin{aligned} C &:= \int_0^{2\pi} u^-(\tau) e^{\alpha\tau} \sinh(\beta(2\pi - \tau)) d\tau, \\ D &:= \int_0^{2\pi} e^{\alpha\tau} u^-(\tau) \omega_0 \sinh(-\beta(2\pi - \tau) + \eta) d\tau. \end{aligned}$$

Now, by direct substitution of the state transition matrices and Jacobian from Lemma 3.15 into Equation (3.22)

$$P_{\mathcal{A}_{\omega_0, \zeta}}(u^-)(t) = u^-(t) + \frac{e^{\alpha(t-2\pi)}}{\beta^2(y_{11}y_{22} - y_{12}y_{21})}$$

$$\begin{aligned}
& \times \begin{bmatrix} -y_{22} \sinh(\beta t) - y_{12} \omega_0 \sinh(\beta t + \eta) \\ y_{21} \sinh(\beta t) + y_{11} \omega_0 \sinh(\beta t + \eta) \end{bmatrix}^T \begin{bmatrix} x_1 - \frac{\beta s_f}{e^{-2\pi\alpha}} \\ x_2 - \frac{\beta v_f}{e^{-2\pi\alpha}} \end{bmatrix}, \\
& = u^-(t) + \frac{e^{\alpha(t-2\pi)}}{\beta^2(y_{11}y_{22} - y_{12}y_{21})} \left(- (y_{22} \sinh(\beta t) + y_{12} \omega_0 \sinh(\beta t + \eta)) \right. \\
& \quad \left. \times \left(x_1 - \frac{\beta s_f}{e^{-2\pi\alpha}} \right) + (y_{21} \sinh(\beta t) + y_{11} \omega_0 \sinh(\beta t + \eta)) \left(x_2 - \frac{\beta v_f}{e^{-2\pi\alpha}} \right) \right).
\end{aligned}$$

where x_i , y_{ij} , $i, j = 1, 2$, are all evaluated at 2π , but not shown for clarity. $\square$

Proof of Lemma 3.17

Given A for the damped harmonic oscillator with $0 < \zeta < 1$ we compute the state transition matrices in (3.38). To find $J_\varphi(0)$ we must solve Equation (3.17) where

$$\tilde{A} = \begin{bmatrix} 0 & 1 & \vdots & 0 & 0 \\ -\omega_0^2 & -2\omega_0\zeta & \vdots & 0 & 0 \\ \hline 0 & 0 & \vdots & 0 & 1 \\ 0 & 0 & \vdots & -\omega_0^2 & -2\omega_0\zeta \end{bmatrix} \quad \tilde{b} = \frac{e^{\alpha t}}{\tilde{\beta}} \begin{bmatrix} 0 \\ \sin(\tilde{\beta}t) \\ \hline 0 \\ -\omega_0 \cos(\tilde{\beta}t + \gamma) \end{bmatrix}.$$

Then using Equation (3.17)

$$y(2\pi) = \int_0^{2\pi} \frac{e^{\alpha\tau}}{\tilde{\beta}} \begin{bmatrix} e^{A(2\pi-\tau)} & \vdots & \mathbf{0}_{2 \times 2} \\ \hline \mathbf{0}_{2 \times 2} & \vdots & e^{A(2\pi-\tau)} \end{bmatrix} \begin{bmatrix} 0 \\ \sin(\tilde{\beta}\tau) \\ \hline 0 \\ -\omega_0 \cos(\tilde{\beta}\tau + \gamma) \end{bmatrix} d\tau.$$

After integration, we have expression (3.39). Recall (as in Lemmas 3.8–3.15)

$$J_\varphi(0) = \begin{bmatrix} y_1(2\pi) & \vdots & y_2(2\pi) \end{bmatrix},$$

which after inverting yields the expression in (3.32). $\square$

Proof of Corollary 3.18

We begin by computing $x(2\pi)$.

$$\begin{aligned}
x(2\pi) &= \frac{e^{-2\pi\alpha}}{\tilde{\beta}} \begin{bmatrix} \omega_0 \cos(2\pi\tilde{\beta} + \gamma) & \sin(2\pi\tilde{\beta}) \\ -\omega_0^2 \sin(2\pi\tilde{\beta}) & \omega_0 \cos(2\pi\tilde{\beta} - \gamma) \end{bmatrix} \begin{bmatrix} s_0 \\ v_0 \end{bmatrix} + \int_0^{2\pi} \frac{e^{-(2\pi-\tau)\alpha}}{\tilde{\beta}} \\
& \quad \times \begin{bmatrix} \omega_0 \cos((2\pi-\tau)\tilde{\beta} + \gamma) & \sin((2\pi-\tau)\tilde{\beta}) \\ -\omega_0^2 \sin((2\pi-\tau)\tilde{\beta}) & \omega_0 \cos((2\pi-\tau)\tilde{\beta} - \gamma) \end{bmatrix} \begin{bmatrix} 0 \\ u^-(\tau) \end{bmatrix} d\tau
\end{aligned}$$

$$= \frac{e^{-2\pi\alpha}}{\tilde{\beta}} \begin{bmatrix} s_0\omega_0 \cos(2\pi\tilde{\beta} + \gamma) + v_0 \sin(2\pi\tilde{\beta}) + C \\ -s_0\omega_0^2 \sin(2\pi\tilde{\beta}) + v_0\omega_0 \cos(2\pi\tilde{\beta} - \gamma) + D \end{bmatrix}$$

where

$$C := \int_0^{2\pi} e^{\alpha\tau} \sin(\tilde{\beta}(2\pi - \tau)) u^-(\tau) d\tau,$$

$$D := \int_0^{2\pi} e^{\alpha\tau} \omega_0 \cos((2\pi - \tau)\tilde{\beta} + \gamma) u^-(\tau) d\tau.$$

In the following we omit the arguments of x and y to save space. After direct substitution of the results from Lemma 3.17 into Equation (3.22) and simple matrix multiplication one finds

$$\begin{aligned} P_{\mathcal{A}_{\omega_0, \zeta}}(u^-)(t) &= u^-(t) + \frac{e^{\alpha t}}{\tilde{\beta}^2(y_{11}y_{22} - y_{12}y_{21})} \\ &\quad \times \begin{bmatrix} -y_{22} \sin(\tilde{\beta}t) - y_{12}\omega_0 \cos(\tilde{\beta}t + \gamma) \\ y_{21} \sin(\tilde{\beta}t) + y_{11}\omega_0 \cos(\tilde{\beta}t + \gamma) \end{bmatrix}^T \begin{bmatrix} x_1 e^{2\pi\alpha} - \tilde{\beta}s_f \\ x_2 e^{2\pi\alpha} - \tilde{\beta}v_f \end{bmatrix}, \\ &= u^-(t) + \frac{e^{\alpha(t-2\pi)}}{\tilde{\beta}^2(y_{11}y_{22} - y_{12}y_{21})} \left(- \left(y_{22} \sin(\tilde{\beta}t) + y_{12}\omega_0 \cos(\tilde{\beta}t + \gamma) \right) \right. \\ &\quad \left. \times \left(x_1 - \frac{\tilde{\beta}s_f}{e^{-2\pi\alpha}} \right) + \left(y_{21} \sin(\tilde{\beta}t) + y_{11}\omega_0 \cos(\tilde{\beta}t + \gamma) \right) \left(x_2 - \frac{\tilde{\beta}v_f}{e^{-2\pi\alpha}} \right) \right). \end{aligned}$$

□

Data availability. The full resolution Matlab graph/plot files that support the findings of this study are available from the corresponding author upon request.

Conflict of interest. The authors have no competing, or conflict of, interests to declare that are relevant to the content of this article.

Acknowledgements. The authors offer their warm thanks to William Hager who made useful comments on an earlier preprint version of their paper in [40]. They are grateful to Walaa Moursi for pointing to a specific result in [29] about convergence of the Douglas–Rachford algorithm. BIC was supported by an Australian Government Research Training Program Scholarship. No funding was received by RSB and CYK to assist with the preparation of this manuscript.

REFERENCES

- [1] H.M. Amman and D.A. Kendrick, Computing the steady state of linear quadratic optimization models with rational expectations. *Econ. Lett.* **58** (1998) 185–191.
- [2] R.S. Burachik, C.Y. Kaya and S.N. Majeed, A duality approach for solving control-constrained linear-quadratic optimal control problems. *SIAM J. Control Optim.* **52** (2014) 1771–1782.
- [3] C. Büskens and 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.
- [4] B. Christiansen, H. Maurer and O. Zirn, Optimal control of machine tool manipulators, in edited by M. Diehl, Recent Advances in Optimization and its Applications in Engineering. Springer-Verlag, Berlin, Heidelberg (2010) 451–460.

- [5] B. Kugelman and H.J. Pesch, New general guidance method in constrained optimal control. I. Numerical method. *J. Optim. Theory Appl.* **67** (1990) 421–435.
- [6] H. Maurer and H.J. Oberle, Second order sufficient conditions for optimal control problems with free final time: The Riccati approach. *SIAM J. Control Optim.* **41** (2003) 380–403.
- [7] T. Mouktonglang, Innate immune response via perturbed LQ-control problem. *Adv. Stud. Biol.* **3** (2011) 327–332.
- [8] M. Athans and P. Falb, Optimal Control: An Introduction to the Theory and Its Applications. McGraw-Hill, Inc., New York (1966).
- [9] D.E. Kirk, Optimal Control Theory: An Introduction. Prentice-Hall, Inc., New Jersey (1970).
- [10] J. Klamka, Controllability and Minimum Energy Control. Springer, Cham, Switzerland (2019).
- [11] S.P. Sethi, Optimal Control Theory: Applications to Management Science and Economics, 1st edn. Springer, Cham, Switzerland (2019).
- [12] H.H. Bauschke and V.R. Koch, Projection Methods: Swiss Army Knives for Solving Feasibility and Best Approximation problems with Halftspaces. Infinite Products of Operators and Their Applications (2012) 1–40.
- [13] F.J. Aragón Artacho, J.M. Borwein and M.K. Tam, Douglas–Rachford feasibility methods for matrix completion problems. *ANZIAM J.* **55** (2014) 299–326.
- [14] S. Gravel and V. Elser, Divide and concur: a general approach to constraint satisfaction. *Phys. Rev. E* **78** (2008) 036706.
- [15] H.H. Bauschke, 8 Queens, Sudoku, and Projection Methods. <https://carma.newcastle.edu.au/resources/jon/Preprints/Books/CUP/Material/Lions-Mercier/Heinz.Bauschke.pdf>, 2008.
- [16] F.J. Aragón Artacho, R. Campoy and V. Elser, An enhanced formulation for solving graph coloring problems with the Douglas–Rachford algorithm. *J. Glob. Optim.* **77** (2020) 383–403.
- [17] Y. Censor, M.D. Altschuler and W.D. Powlis, On the use of Cimminos simultaneous projections method for computing a solution of the inverse problem in radiation therapy treatment planning. *Inverse Probl.* **4** (1988) 607–623.
- [18] B. O’Donoghue, G. Stathopoulos and S. Boyd, A splitting method for optimal control. *IEEE Trans. Contr. Syst. Tech.* **21** (2013) 2432–2442.
- [19] H.H. Bauschke, R.S. Burachik and C.Y. Kaya, Constraint splitting and projection methods for optimal control of double integrator, in Splitting Algorithms, Modern Operator Theory, and Applications. Springer (2019) 45–68.
- [20] R. Fourer, D.M. Gay and B.W. Kernighan, AMPL: A Modeling Language for Math. Progr., 2nd edn. Brooks/Cole Publishing Company/Cengage Learning (2003).
- [21] A. Wächter and L.T. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math. Progr.* **106** (2006) 25–57.
- [22] V.G. Boltyanskii, R.V. Gamkrelidze, E.F. Mishchenko and L.S. Pontryagin, The Mathematical Theory of Optimal Processes. John Wiley & Sons, New York (1962).
- [23] M.R. Hestenes, Calculus of Variations and Optimal Control Theory. John Wiley & Sons, New York (1966).
- [24] R.B. Vinter, Optimal Control. Birkhäuser, Boston (2000).
- [25] B.S. Mordukhovich, Variational Analysis and Generalized Differentiation II: Applications. Springer-Verlag, Berlin, Heidelberg (2006).
- [26] F. Clarke, Functional Analysis, Calculus of Variations and Optimal Control. Springer-Verlag, London (2013).
- [27] W.J. Rugh, Linear System Theory, 2nd edn. Pearson (1995).
- [28] R.L. Borrelli and C.S. Coleman, Differential Equations: A Modeling Perspective, 2nd edn. John Wiley and Sons (2004).
- [29] H.H. Bauschke and P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces. 2nd edn. Springer (2017).
- [30] J. Douglas and H.H. Rachford, On the numerical solution of heat conduction problems in two and three space variables. *Trans. Am. Math. Soc.* **82** (1956) 421–439.
- [31] P.-L. Lions and B. Mercier, Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16** (1979) 964–979.
- [32] J. Eckstein and D.P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Progr., Ser. A* **55** (1992) 293–318.
- [33] B.F. Svaiter, On weak convergence of the Douglas-Rachford method. *SIAM J. Control Optim.* **49** (2011) 280–287.
- [34] H.H. Bauschke and W.M. Moursi, On the Douglas–Rachford algorithm. *Math. Program., Ser. A* **164** (2017) 263–284.
- [35] M.M. Alves, J. Eckstein, M. Geremia and J.G. Melo, Relative-error inertial-relaxed inexact versions of Douglas–Rachford and ADMM splitting algorithms. *Comput. Optim. Appl.* **75** (2020) 389–422.
- [36] B.F. Svaiter, A weakly convergent fully inexact Douglas–Rachford method with relative error tolerance. *ESAIM. Control Optim. Calc. Var.* **25** (2019).
- [37] M.M. Alves and M. Geremia, Iteration complexity of an inexact Douglas–Rachford method and of a Douglas–RachfordTsengs F–B four-operator splitting method for solving monotone inclusions. *Numer. Algorithms* **82** (2019) 263–295.
- [38] H.H. Bauschke and W.M. Moursi, On the Douglas–Rachford algorithm for solving possibly inconsistent optimization problems. *Math. Oper. Res.* <https://doi.org/10.1287/moor.2022.1347>.
- [39] W.W. Hager, Runge–Kutta methods in optimal control and the transformed adjoint system. *Numer. Math.* **87** (2000) 247–282.
- [40] R.S. Burachik, B.I. Caldwell and C.Y. Kaya, Projection methods for control-constrained minimum-energy control problems. [arXiv:2210.17279v1](https://arxiv.org/abs/2210.17279v1), <https://arxiv.org/abs/2210.17279>.
- [41] H.H. Bauschke and J.M. Borwein, On projection algorithms for solving convex feasibility problems. *SIAM Rev.* **38** (1996) 367426.

- [42] J. von Neumann, On rings of operators. Reduction theory. *Ann. Math.* **50** (1949) 401–485.
- [43] J.P. Boyle and R.L. Dykstra, A method for finding projections onto the intersection of convex sets in Hilbert spaces, in *Advances in Order Restricted Statistical Inference*, Vol. 37 of *Lecture Notes in Statistics*. Springer (1986) 28–47.
- [44] F.J. Aragón Artacho and R. Campoy, A new projection method for finding the closest point in the intersection of convex sets. *Comput. Optim. Appl.* **69** (2018) 99–132.



Please help to maintain this journal in open access!

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at <https://edpsciences.org/en/subscribe-to-open-s2o>.