

## A PRIMAL–DUAL ALGORITHM AS APPLIED TO OPTIMAL CONTROL PROBLEMS

REGINA S. BURACHIK, C. YALÇIN KAYA, AND XUEMEI LIU

**ABSTRACT.** We use a primal–dual technique for solving infinite dimensional problems arising from optimal control. Namely, we solve as examples a control-constrained double integrator optimal control problem and the challenging control-constrained free flying robot optimal control problem by means of the primal–dual scheme. The algorithm we use is an epsilon-subgradient method that can also be interpreted as a penalty function method. We provide extensive comparisons of our approach with a traditional numerical approach.

### 1. INTRODUCTION

Let  $U$  be a reflexive Banach space,  $H$  a Hilbert space and let  $K \subset U$ . Consider the following infinite dimensional equality constrained optimization problem:

$$(P) \quad \min_{u \in U} \varphi(u) \text{ s.t. } u \in K, \quad h(u) = 0,$$

where  $\varphi : U \rightarrow \mathbb{R} \cup \{\infty\}$  is lower semi-continuous and  $h : U \rightarrow H$  is a given function. In general, (P) is not convex, so if we are to use duality to solve it, we need to use an augmented Lagrangian approach. The type of Lagrangian we consider for addressing Problem (P) is an extension of the one in [7, 11] to infinite dimensions. The infinite dimensional case was recently studied in [14], where Problem (P) is addressed via a primal–dual scheme where the augmented Lagrangian  $\ell : U \times \mathbb{R}_+ \rightarrow \mathbb{R}$  is defined as follows

$$(1.1) \quad \ell(u, c) := \inf_{v \in K} \varphi(v) - \langle u, A(h(v)) \rangle + c \sigma(h(v)),$$

where  $u \in U$ ,  $c \geq 0$ ,  $A : H \rightarrow H$ , is a suitable map, and  $\sigma : H \rightarrow \mathbb{R}_+$  verifies  $\sigma(v) = 0$  if and only if  $v = 0$ . The resulting primal–dual scheme is paired in [14] with an epsilon subgradient technique that solves the dual problem. An important advantage of (1.1) is that, unlike other available versions (such as the ones proposed in [8–10, 12, 24]), the use of this  $\ell$  induces a penalty function method for the choice  $A = 0$ . The aim of the present paper is to exploit this feature, and to use it for solving optimal control problems. More precisely, we will take  $A = 0$  and  $\sigma$  as a suitable norm in  $H$ . Since it can be seen as a penalty approach, we denote the resulting epsilon-subgradient method as a primal–dual penalty (PDP) algorithm.

---

2020 *Mathematics Subject Classification.* 49M29, 49M30, 49M37.

*Key words and phrases.* Optimal control, augmented Lagrangian, Banach space, nonconvex optimization, nonsmooth optimization, subgradient methods, duality scheme, penalty function methods.

The PDP algorithm uses a subgradient direction as its update rule (for the penalty parameter) and improves the dual values in each iteration.

The classical advantages of the augmented Lagrangian scheme here are that, (i) even when the original problem is not convex, the dual problem is, and hence it can be solved by standard techniques from convex analysis, (ii) there is no gap between primal and dual optimal values, and (iii) solving the dual problem provides a primal solution.

The fact that the primal–dual approach using the augmented Lagrangian (1.1) enjoys the advantages listed in (i)–(iii) has been proved in [14]. The interested reader can also see [21, 22] for an excellent introduction on different types of Lagrangians and their applications in solving various kinds of problems. We note that a different type of augmented Lagrangian technique has been used for solving finite-dimensional problems in [13].

Our first aim is theoretical, and it consists of determining a wide enough family of problems (P) such that the PDP algorithm is well defined when applied to (P). This is needed to ensure that properties (i)–(iii) will hold for our scheme. We establish this fact in Theorem 3.3.

Our second aim is practical, and it is to show that we can apply the PDP algorithm to solve challenging optimal control problems. We achieve this aim by addressing two optimal control problems that do not have an analytical solution available. These problems are the control-constrained double integrator and the free flying robot [2, 3, 28, 29]. While the first one of these problems is convex, the second one is highly nonconvex.

Projection-type methods can be used to solve some optimal control problems (for example the one in [4]), as long as they are convex. Our approach via the PDP algorithm, however, can solve also non-convex instances of these problems. Even when dealing with convex problems, the projection is usually difficult to compute unless we project onto simple sets. When compared with the penalty methods proposed in [17–20], we note that the latter works consider either simple problems or those with analytical solutions. Moreover, the methods in [17–20] have not been implemented.

The paper is organized as follows. In Section 2, we give the preliminaries on functional analysis, which help in building our assumptions on Problem (P). Section 3 provides our theoretical framework, where we show that the family of problems we address verifies the necessary assumptions (see Theorem 3.3). In this section we recall (i) the properties of the duality framework, as well as (ii) the definition of the PDP method and its properties, which were established in [14]. In Section 4, we define a class of optimal control problems in the format of Problem (P), and derive its conditions of optimality. Also in this section we describe our discretization scheme. In Sections 5 and 6, we implement the PDP algorithm for solving the constrained double integrator, and the challenging free-flying robot problem, respectively. In these sections we also show the performance of our approach and compare it with a conventional numerical approach. Section 7 contains the conclusion and further discussion. To simplify the presentation, longer, or more involved, proofs are given in an Appendix at the end of our paper.

2. PRELIMINARIES

To determine a general family of problems and show that certain optimal control problems belong to that family, we need to recall a few results from functional analysis, which we do in the next subsection.

**2.1. Some Functional Analysis Tools.** Let  $X$  be a reflexive Banach space,  $X^*$  its topological dual (i.e., the set all continuous linear functionals from  $X$  to  $\mathbb{R}$ ), and  $H$  a Hilbert space. We denote by  $\langle \cdot, \cdot \rangle$  both the duality product in  $X \times X^*$  and the scalar product in  $H$ . Unless explicitly indicated, we denote by  $\|\cdot\|$  the norm of  $X$  or  $H$ . We use the notation  $\mathbb{R}_{+\infty} = \mathbb{R}_\infty := \mathbb{R} \cup \{+\infty\}$ . Given a function  $g : X \rightarrow \mathbb{R}_{+\infty} \cup \{-\infty\}$ , the *effective domain* of  $g$  is the set  $\text{dom } g := \{x \in X : g(x) < +\infty\}$ . We say that  $g$  is *proper* if  $g(x) > -\infty$  and  $\text{dom } g \neq \emptyset$ . The set  $\text{lev}_g(\alpha) := \{x \in X : g(x) \leq \alpha\}$  is the  $\alpha$ -*level set* of  $g$ . Given  $C \subset X$ , the *indicator function* of  $C$  is defined as  $\delta_C(v) := 0$  if  $v \in C$  and  $+\infty$  otherwise. If  $C = \{z\}$  is a singleton, we denote  $\delta_{\{z\}} =: \delta_z$ .

The topology induced by the norm (in  $X$  or  $H$ ), is called the *strong* topology. The weak topology in  $X$  (weak topology in  $H$ ) is the coarsest topology that makes all elements of  $X^*$  (all elements of  $H^* = H$ ) continuous. We will need the following definitions concerning the weak topology. For  $A \subset X$ , we denote by  $\text{cl } A$  the strong closure of  $A$  and by  $\overline{A}^w$  the weak closure of  $A$ . In most of what follows, when a topological property is mentioned by its own, this means that the property holds w.r.t. the strong (i.e., the norm) topology. For instance, if we write “ $A$  is closed”, we mean “ $A$  is STRONGLY closed”. If a property holds w.r.t. the weak topology, we will mention the term “weak” (or “weakly”) explicitly. For instance, we may say weakly closed, (or w-closed), weakly compact (or w-compact), etc. Recall that a function  $\varphi : X \rightarrow \mathbb{R}_{+\infty}$  is *weakly lower semi-continuous* (*w-lsc*) when it is lsc w.r.t. the weak topology in  $X$ . Namely, when *epi*  $\varphi$  is w-closed. Given  $\{u_n\}, \{w_n\} \subset X$ , we denote the fact that  $\{u_n\}$  converges weakly to  $u$  as  $u_n \rightharpoonup u$ , and the fact that  $\{w_n\}$  converges strongly to  $w$  as  $w_n \rightarrow w$ .

We recall next some well-known facts from functional analysis, most of which can be found in [6]. The reader familiar with functional analysis can skip this section, with the exception of Lemma 2.10, which, to our knowledge, is new.

**Fact 2.1.** Let  $X$  be a reflexive Banach space,  $H$  be a Hilbert space. Assume that  $K \subset X$  is nonempty. The following hold.

- (a)  $K \neq \emptyset$  is weakly closed if and only if the indicator function  $\delta_K$  is proper and w-lsc.
- (b) If  $K \subset X$  is weakly compact, then it is weakly closed.
- (c) A convex subset of  $X$  is weakly closed if and only if it is closed.

We recall the following definitions.

**Definition 2.1** (Weak compactness; sequential compactness; coercive).

Let  $X$  be a Banach space,  $A \subset X$  and  $\varphi : X \rightarrow \mathbb{R}_{+\infty}$ .

- (a)  $A$  is *weakly-compact* when its weak closure,  $\overline{A}^w$ , is compact w.r.t the weak topology.
- (b)  $A \subset X$  is *sequentially-compact* (respectively, *weakly sequentially-compact*) when every  $\{x_n\} \subset A$  has a subsequence converging strongly (respectively, weakly) to a limit in  $A$ .

(c) The function  $\varphi : X \rightarrow \mathbb{R}_{+\infty}$  is *coercive* when  $\lim_{\|x\| \rightarrow \infty} \varphi(x) = +\infty$ .

The equivalence between compactness and sequential-compactness in normed spaces allows the use of sequences when dealing with compact sets. To deal with weakly compact sets in terms of sequences, we will use Eberlein–Smulian theorem [6, Problem 10(3), p. 448], recalled next.

**Theorem 2.2** (Eberlein–Smulian). *Let  $A \subset X$ . Then  $A$  is weakly compact if and only if it is weakly sequentially-compact.*

Next we quote results that connect boundedness, closedness and compactness both in strong and weak topologies. The next result, a corollary of Bourbaki–Alaoglu’s theorem, is [6, Corollary 3.22].

**Theorem 2.3.** *Let  $E$  be a reflexive Banach space. Let  $K \subset E$  be a bounded, closed, and convex subset of  $E$ . Then  $K$  is weakly compact.*

**Corollary 2.4.** *If  $X$  is a Banach space, then every weakly compact set is closed and bounded. Consequently, every weakly convergent sequence must be bounded.*

We will also need the following two results involving functions defined on  $X$ . The first one is [6, Corollary 3.9], and is a direct consequence of Fact 2.1(c). The second result can be found, e.g., in [14, Corollary 2.2].

**Theorem 2.5.** *Assume that  $\varphi : X \rightarrow \mathbb{R}_{+\infty}$  is convex. Then  $\varphi$  is *w-lsc* if and only if it is *lsc*.*

**Corollary 2.6.** *Let  $X$  be a reflexive Banach space and let  $\varphi : X \rightarrow \mathbb{R}_{+\infty}$  be *w-lsc*. Then  $\varphi$  is *coercive* if and only if all its level sets are weakly compact. In this situation, all the level sets are closed and bounded.*

**Definition 2.7.** Let  $1 \leq p < \infty$ ,  $\mathcal{L}^p([0, t_f]; \mathbb{R}^m)$  be the Banach space of Lebesgue measurable functions  $z : [0, t_f] \rightarrow \mathbb{R}^m$ , with finite  $\mathcal{L}^p$  norm, denoted  $\|\cdot\|_{\mathcal{L}^p}$ , namely,

$$\mathcal{L}^p([0, t_f]; \mathbb{R}^m) = \left\{ z : [0, t_f] \rightarrow \mathbb{R}^m \mid \|z\|_{\mathcal{L}^p} = \left( \int_0^{t_f} \|z(t)\|_p^p dt \right)^{1/p} < \infty \right\},$$

where  $\|\cdot\|_p$  is the  $\ell_p$  norm in  $\mathbb{R}^m$ .  $\mathcal{W}^{1,2}([0, t_f]; \mathbb{R}^m)$  is the Sobolev space of absolutely continuous functions, namely,

$$\mathcal{W}^{1,2}([0, t_f]; \mathbb{R}^m) = \{ z \in \mathcal{L}^2([0, t_f]; \mathbb{R}^m) \mid \dot{z} = dz/dt \in \mathcal{L}^2([0, t_f]; \mathbb{R}^m) \},$$

endowed with the norm

$$\|z\|_{\mathcal{W}^{1,2}} = (\|z\|_2^2 + \|\dot{z}\|_2^2)^{1/2}.$$

We will make use of the following result, which is [6, Proposition 3.5(iv)].

**Proposition 2.8.** *Let  $X$  be Banach space. Consider two sequences  $\{u_n\} \subset X$ ,  $\{w_n\} \subset X^*$  and let  $u \in X$ ,  $w \in X^*$  be such that  $u_n \rightharpoonup u$  (i.e.,  $\{u_n\}$  converges weakly to  $u$ ), and  $w_n \rightarrow w$  (i.e.,  $\{w_n\}$  converges strongly to  $w$ ). Then*

$$\lim_{n \rightarrow \infty} \langle u_n, w_n \rangle = \langle u, w \rangle.$$

The next result follows from Lebesgue’s Dominated Convergence Theorem (see, e.g., [6, Theorem 4.2]).

**Theorem 2.9.** Let  $f_n : [0, T] \rightarrow \mathbb{R}$  for every  $n \in \mathbb{N}$  verifying that  $\lim_{n \rightarrow \infty} f_n(t) \in \mathbb{R}$  for every  $t \in [0, T]$ . Assume that there exists  $M > 0$  such that  $|f_n(t)| \leq M$  for all  $t \in [0, T]$ . Define  $f : [0, T] \rightarrow \mathbb{R}$  such that  $f(t) := \lim_{n \rightarrow \infty} f_n(t)$  for all  $t \in [0, T]$ . Then,

(a)  $f \in \mathcal{L}^1([0, T]; \mathbb{R})$ .

(b) For every  $t \in [0, T]$  we have  $\lim_{n \rightarrow \infty} \int_0^t f_n(s) ds = \int_0^t f(s) ds$ .

To study the ODE systems of sections 5 and 6, we will need the lemma below. This lemma establishes the sequential weak continuity of a family of real-valued functions defined in  $(\mathcal{L}^2([0, T]; \mathbb{R}))^m$ .

**Lemma 2.10.** Let  $m \in \mathbb{N}$ ,  $T > 0$  and  $u := (u^1, \dots, u^m) \in (\mathcal{L}^2([0, T]; \mathbb{R}))^m$ . Assume that

- (a)  $\varphi : \mathbb{R} \rightarrow \mathbb{R}$  is continuous and globally bounded (i.e., exists  $L_1 > 0$  s.t.  $|\varphi(t)| \leq L_1$  for all  $t \in \mathbb{R}$ ),  
 (b) Fix  $a, b_1, \dots, b_m \in \mathbb{R}$ , and define  $\pi(u) : [0, T] \rightarrow \mathbb{R}$  as

$$\pi(u)(t) := a + \sum_{j=1}^m b_j \int_0^t \left( \int_0^r u^j(s) ds \right) dr.$$

- (c) Fix  $r, t \in [0, T]$  s.t.  $r \leq t$  and  $j \in \{1, \dots, m\}$ . Define  $\eta_j^\varphi(\cdot, r), \rho_j^\varphi(\cdot, t) : (\mathcal{L}^2([0, T]; \mathbb{R}))^m \rightarrow \mathbb{R}$  as

$$\eta_j^\varphi(u, r) := \int_0^r u^j(s) \varphi(\pi(u)(s)) ds,$$

$$\rho_j^\varphi(u, t) := \int_0^t \left( \int_0^r u^j(s) \varphi(\pi(u)(s)) ds \right) dr = \int_0^t \eta_j^\varphi(u, r) dr$$

If  $\{u_k\} \subset (\mathcal{L}^2([0, T]; \mathbb{R}))^m$  and  $u_k \rightharpoonup u$ , then we have

$$(2.1) \quad \lim_{k \rightarrow \infty} \eta_j^\varphi(u_k, r) = \eta_j^\varphi(u, r), \forall r \in [0, T]$$

and

$$(2.2) \quad \lim_{k \rightarrow \infty} \rho_j^\varphi(u_k, t) = \rho_j^\varphi(u, t), \forall t \in [0, T]$$

Namely, the functions  $\eta_j^\varphi(\cdot, r)$  and  $\rho_j^\varphi(\cdot, t)$  are  $w$ -sequentially continuous for every  $r, t \in [0, T]$ .

*Proof.* See the proof of Lemma 2.10 in Appendix. □

### 3. PRIMAL AND DUAL PROBLEMS

**3.1. Theoretical Framework for Duality.** Following [15, Section 2.2], we embed Problem (P) into a family of parametrized problems using a function that coincides with  $\varphi$  when the parameter is zero. This tool is given next.

**Definition 3.1.** A dualizing parameterization for (P) is a function  $f : U \times H \rightarrow \bar{\mathbb{R}}$  that verifies  $f(u, 0) = \varphi(u)$  for all  $u \in U$ .

The next definition is [15, Definition 5.1] and will be our basic assumption for the dualizing parametrization.

**Definition 3.2.** A function  $f : U \times H \rightarrow \bar{\mathbb{R}}$  is said to be *weakly level-compact* if for each  $\bar{z} \in H$  and  $\alpha \in \mathbb{R}$  there exist a weakly open neighbourhood  $V \subset H$  of  $\bar{z}$ , and a weakly compact set  $B \subset U$ , such that

$$\text{lev}_{z,f}(\alpha) := \{u \in U : f(u, z) \leq \alpha\} \subset B \text{ for all } z \in V.$$

We next list the basic assumptions of the primal–dual framework.

- (H0) The objective function  $\varphi : U \rightarrow \mathbb{R}_\infty$  is proper and w-lsc.
- (H1) The function  $\varphi$  has weakly compact level sets.
- (H2) The dualizing parameterization  $f$  is proper (i.e.,  $\text{dom } f \neq \emptyset$  and  $f(u, z) > -\infty, \forall (u, z) \in U \times H$ ), w-lsc and weakly level-compact (see Definition 3.2).

The result below will be used in our application to optimal control problems.

**Theorem 3.3** (Problem (P) verifies (H0)–(H2)). *Let  $U$  be a reflexive Banach space and let  $H$  be a Hilbert space. Consider a function  $\varphi : U \rightarrow \mathbb{R}_{+\infty}$ , a set  $K \subset U$ , and a function  $h : U \rightarrow H$ . Consider Problem (P), i.e.,*

$$(P) \quad \min \varphi(u) \text{ s.t. } u \in K \text{ and } h(u) = 0.$$

*Assume that  $S(P)$ , the solution set of Problem (P), is nonempty and that the following hold.*

- (a) *The objective function  $\varphi$  is proper, coercive and w-lsc.*
- (b) *For every  $z \in H$ , the set  $K \cap h^{-1}(z)$  is weakly closed, where  $h^{-1}(z) := \{u \in U : h(u) = z\}$ .*
- (c) *The dualizing parameterization  $f : U \times H \rightarrow \mathbb{R}_\infty$  is defined as follows*

$$f(u, z) := \varphi(u) + \delta_K(u) + \delta_z(h(u)).$$

*Then Problem (P) verifies assumptions (H0)–(H2).*

*Proof.* We note that (H0) holds automatically by the choice of  $\varphi$  in (a). To check (H1), we need to show that all level sets of  $\varphi$  are weakly compact. By (a)  $\varphi$  is coercive and w-lsc, so we can apply Corollary 2.6 to conclude that all its level sets are weakly compact. Thus (H1) holds. We proceed to check (H2). From the definition of indicator function we have that

$$f(u, z) = \varphi(u) + \delta_{K \cap h^{-1}(z)}(u) \geq \varphi(u) > -\infty.$$

Moreover,  $f(u, z)$  is not identically  $+\infty$  because  $\emptyset \neq S(P) \subset K \cap h^{-1}(0)$ . Hence  $f$  is proper. We proceed now to show that  $f$  is w-lsc. Indeed, it is enough to show that it is the sum of w-lsc functions. By (a),  $\varphi$  is w-lsc. Assumption (b) and Fact 2.1(a) imply that  $\delta_{K \cap h^{-1}(z)}$  is weakly-lsc, too. Altogether,  $f$  is the sum of w-lsc functions and hence w-lsc. We proceed now to show that  $f$  is weakly level compact. Fix  $z_0 \in H$  and  $W$  any weakly open set containing  $z_0$ . With the notation of Definition 3.2, use assumption (c) to write for any  $z \in W$

$$\begin{aligned} \text{lev}_{z,f}(\alpha) &= \{u \in U : f(u, z) \leq \alpha\} \\ &= \{u \in K : \varphi(u) \leq \alpha, h(u) = z\} \subset \{u \in K : \varphi(u) \leq \alpha\} \subset \text{lev}_\varphi(\alpha), \end{aligned}$$

where  $lev_\varphi(\alpha)$  denotes the level set of  $\varphi$ . Since  $\varphi$  is weakly level compact,  $lev_\varphi(\alpha) =: B$  is w-compact. Since  $B$  does not depend on  $z$ , the expression above yields

$$\bigcup_{z \in W} lev_{z,f}(\alpha) \subset B,$$

which implies that  $f$  is weakly level compact. Therefore (H2) holds. □

**Remark 3.4.** Problem (P) as in Theorem 3.3 has been considered in [10, Example 2.1], where  $h$  is assumed to have a weakly closed graph. In infinite dimensions, this assumption may be too restrictive or hard to establish. We replace it here by the less restrictive assumption (b) which is enough to ensure the w-lsc of the duality parametrization required for (H2) to hold. In later sections, we will show that (a)-(c) in Theorem 3.3 hold for our optimal control examples.

We define next the augmented Lagrangian function, and the resulting problem dual to (P). As mentioned in the introduction, this Lagrangian is a particular case of that analyzed in [14, Section 3]. Namely, we take  $A = 0$  and  $\sigma$  a suitable norm in (1.1).

**Definition 3.5** (Augmented Lagrangian and associated dual problem). Let  $U, H$  and  $h$  be as in Problem (P). Define  $K_0 := K \cap \{u \in U : h(u) = 0\}$ , i.e.,  $K_0$  is the constraint set for Problem (P). Take  $\|\cdot\|$  a norm in  $H$ . Assume that the dualizing parameterization  $f$  is given by

$$(3.1) \quad f(u, z) := \varphi(u) + \delta_K(u) + \delta_z(h(u)),$$

and that it satisfies assumption (H2). We consider the following type of augmented Lagrangian  $\ell : U \times \mathbb{R}_+ \rightarrow \mathbb{R}_{-\infty}$  for Problem (P):

$$(3.2) \quad \ell(u, c) := \inf_{z \in H} \{f(u, z) + c\|z\|\} = \begin{cases} \{\varphi(u) + c\|h(u)\|\}, & \text{if } u \in K, \\ +\infty & \text{otherwise.} \end{cases}$$

The resulting dual function  $q : \mathbb{R}_+ \rightarrow \mathbb{R}_{-\infty}$  is

$$(3.3) \quad q(c) := \inf_{u \in U} \ell(u, c) = \inf_{u \in K} \varphi(u) + c\|h(u)\|,$$

where we used (3.1) and (3.2) in the last equality. The dual problem of (P) is given by

$$(D) \quad \text{maximize } q(c) \text{ s.t. } c \geq 0.$$

Denote by  $M_P := \inf_{u \in K_0} \varphi(u)$  and by  $M_D := \sup_{c \geq 0} q(c)$  the optimal values of the primal and dual problem, respectively. The primal and dual solution sets are denoted by  $S(P)$  and  $S(D)$ , respectively.

**Remark 3.6** (Finite primal value for Problem (P)). Assumption (H0) implies that  $\varphi$  is proper, so Definition 3.1 yields  $M_P < +\infty$ .

**Remark 3.7** (Optimal control examples). In our examples, we will always have that  $H$ , the co-domain of the function  $h$ , is a finite dimensional Hilbert space. Namely, we will always have that  $H := \mathbb{R}^m$  for some  $m \in \mathbb{N}$ . This will allow us to take the (finite dimensional)  $\ell_1$  norm as the norm in (3.1) and (3.2).

**3.2. Properties of the Primal–Dual Setting.** We next present some basic properties of the dual function given in Definition 3.5. The proof of the proposition below is standard, and can be found in [14].

**Proposition 3.8** (Properties of the dual function). *Let  $q : \mathbb{R}_+ \rightarrow \mathbb{R}_{-\infty}$  be the dual function defined in (3.3). The following facts hold.*

- (i) *The dual function  $q$  is concave, increasing and weakly upper-semicontinuous (i.e.,  $-q$  is w-lsc).*
- (ii) *If  $c_1 \in S(D)$ , then  $c \in S(D)$  for all  $c \geq c_1$ .*

We state next, adapted to our type of Lagrangian, several properties of the primal dual scheme. We start with strong duality, proved in [14, Theorem 3.1].

**Theorem 3.9** (Strong duality for (P)–(D) framework). *Consider the primal–dual problems (P) and (D). Assume that (H0)–(H2) hold. Suppose that there exists some  $\bar{c} \in \mathbb{R}_+$  such that  $q(\bar{c}) > -\infty$ . Then the zero-duality-gap property holds, i.e.  $M_P = M_D$ .*

**Definition 3.10** (Superdifferential of a concave function). Let  $g : \mathbb{R} \rightarrow \mathbb{R}_{-\infty}$  be a concave function. The superdifferential of  $g$  at  $c_0 \in \text{dom}(g) := \{c \in \mathbb{R} : g(c) > -\infty\}$  is the set  $\partial g(c_0)$  defined by

$$\partial g(c_0) := \{v \in \mathbb{R} : g(c) \leq g(c_0) + \langle v, c - c_0 \rangle, \forall c \in \mathbb{R}\}.$$

**Definition 3.11** (Approximations for the primal–dual and Lagrangian). Define the set

$$X(c) := \{u \in U : \varphi(u) + c\|h(u)\| = q(c)\}.$$

Namely,  $X(c)$  is the set of minimizers of the augmented Lagrangian.

The next proposition will be used to justify the stopping criterion in the PDP algorithm.

**Proposition 3.12** (Search direction and stopping criterion for the PDP algorithm).

*Assume that (H0)–(H2) hold for Problem (P). If  $\hat{u} \in X(\hat{c})$ , then the following facts hold.*

- (i)  $\|h(\hat{u})\| \in \partial q(\hat{c})$ .
- (ii) *If  $h(\hat{u}) = 0$  then  $\hat{u}$  is an optimal primal solution, and  $\hat{c}$  is an optimal dual solution. Conversely, assume  $\hat{c} > 0$  and that either  $\hat{u}$  is an optimal primal solution, or  $\hat{c}$  is an optimal dual solution. Then, we must have  $h(\hat{u}) = 0$ .*

*Proof.* Even though the proof of (i) is standard and can be deduced from [10, Proposition 3.1], we include its proof here for convenience of the reader. To prove (i), use the definition of  $q$  in (3.3) to write, for every  $c \geq 0$ ,

$$\begin{aligned} q(c) &= \inf_{u \in K} \varphi(u) + c\|h(u)\| \leq \varphi(\hat{u}) + c\|h(\hat{u})\| \\ &= \varphi(\hat{u}) + \hat{c}\|h(\hat{u})\| + (c - \hat{c})\|h(\hat{u})\| = q(\hat{u}) + (c - \hat{c})\|h(\hat{u})\|, \end{aligned}$$

where we used the fact that  $\hat{u} \in X(\hat{c})$  and the definition of  $q$  in the last equality. The above expression and Definition 3.10 yield  $\|h(\hat{u})\| \in \partial q(\hat{c})$ , establishing (i). The first statement in part (ii) has a proof similar to the one in [10, Proposition



3.1] and hence omitted. So we prove the second statement in (ii). This statement automatically holds when  $\hat{u}$  is an optimal primal solution because  $\hat{u} \in S(P)$  and therefore it satisfies the equality constraints. We proceed now to prove the statement when  $\hat{c}$  is an optimal dual solution. Theorem 3.9 and Remark 3.6 yield

$$\infty > M_P = M_D = q(\hat{c}) = \varphi(\hat{u}) + \hat{c}\|h(\hat{u})\|,$$

where we used the assumption that  $\hat{c}$  is a dual solution in the second equality, and the fact that  $\hat{u} \in X(\hat{c})$  in the third one. Assume that  $h(\hat{u}) \neq 0$ . By Proposition 3.8(ii), for every  $\lambda > 0$  we have that  $\hat{c} + \lambda \in S(D)$  and hence  $q(\hat{c} + \lambda) = M_D$ . We can write

$$\infty > M_P = M_D = q(\hat{c} + \lambda) = \varphi(\hat{u}) + \hat{c}\|h(\hat{u})\| + \lambda\|h(\hat{u})\|.$$

whose right-hand side tends to infinity for  $\lambda \rightarrow +\infty$ . This contradiction implies that  $h(\hat{u}) = 0$ . □

**3.3. The Primal–Dual Penalty (PDP) Algorithm.** Our Lagrangian is given by equation (3.2) in Definition 3.5, and it gives rise to a classical penalty method. This motivates the name *primal–dual penalty (PDP) algorithm*, described below. We use in this algorithm the notation of Problem (P) and Definition 3.5. By Remark 3.7, we always have  $h(u) = (h_1(u), \dots, h_{m_1}(u)) \in \mathbb{R}^{m_1}$  for some  $m_1 \in \mathbb{N}$ . This allows us to consider finite dimensional norms for  $h(u)$  in the definition of the PDP algorithm. Namely, we use the  $\ell_1$  and the  $\ell_\infty$  norms of  $h(u)$ . Our dual variable is  $c_k \in \mathbb{R}_+$ , while our primal variable is a function  $u_k \in (\mathcal{L}^2([0, t_f]; \mathbb{R}))^{m_2}$  for some  $m_2 \in \mathbb{N}$ .

**Algorithm 3.1. Primal–Dual Penalty (PDP) Algorithm**

Let  $\alpha, \varepsilon > 0$ . Choose a sequence  $\{\alpha_k\} \subset (0, \alpha)$ .

**Step 0.** (Initialization) Choose  $c_0 > 0$  and let  $k := 0$ .

**Step 1.** (Solution of Subproblem and Stopping Criterion)

(a) Find  $u_k \in \operatorname{argmin}_{u \in U} \ell(u, c_k)$ .

(b) If  $\|h(u_k)\|_\infty < \varepsilon$ , stop.

**Step 2.** (Selection of step-size and Update of Dual Variables)

Choose  $s_k > 0$  and set  $\tilde{s}_k = (\alpha_k + 1)s_k$ . Update the penalty parameter by

$$c_{k+1} := c_k + \tilde{s}_k \|h(u_k)\|_1.$$

Set  $k := k + 1$ , go to Step 1.

**Remark 3.13.** Proposition 3.8(i) states that the dual function is non-increasing. Moreover, for  $\{c_k\}$  generated by the PDP algorithm, strict increase of the sequence  $\{q(c_k)\}$  is established in [10, Theorem 3.1]).

**Remark 3.14.** By Proposition 3.12(i), the search direction in Step 2 of the PDP algorithm is a classical subgradient direction for improving  $q$ . Proposition 3.12(ii) justifies the stopping criterion in Step 1(b).

Next we describe two choices for the step-size  $s_k$  and the convergence results for each choice.

**3.4. Algorithm PDP-1.** We consider in this section a step-size as in [10, Algorithm 1]. Take two parameters  $\beta > \eta > 0$ . Let  $u_k$  be as Step 1(a). Consider the step-size

$$(3.4) \quad s_k \in [\eta_k, \beta_k],$$

where  $\eta_k := \min\{\eta, \|h(u_k)\|_2\}$  and  $\beta_k := \max\{\beta, \|h(u_k)\|_1 + \|h(u_k)\|_2\}$ , where  $\|\cdot\|_2$  is the finite dimensional  $\ell_2$  norm. With this choice of  $s_k$ , we denote the PDP algorithm as PDP-1.

**Remark 3.15.** Note that a constant step-size for all iterations is admissible.

The next theorem states the convergence results for PDP-1. The proof of part (a) considers two possible cases, according to whether the dual sequence  $\{c_k\}$  is bounded or not. The case of an unbounded sequence has a proof similar to [10, Theorem 3.2]. The case of bounded dual sequence is slightly different, and can be found in [14, Theorem 4.2]. The proof of part (b) follows directly from the fact that the dual sequence is strictly increasing.

**Theorem 3.16** (Convergence of PDP-1). *Assume that  $M_P = M_D$ . Consider the primal sequence  $\{u_k\}$  generated by PDP-1. Take the parameter sequence  $\{\alpha_k\}$  satisfying  $\alpha_k \geq \bar{\alpha}$  for all  $k$  and some  $\bar{\alpha} > 0$ . The following hold.*

- (a) *The primal sequence  $\{u_k\}$  is bounded, all its weak accumulation points are primal solutions, and  $\{q_k\}$  converges to the optimal value  $M_P$ .*
- (b) *If PDP-1 generates an infinite sequence  $\{c_k\}$ , then it converges if and only if it is bounded above, and in this case its supremum is a dual solution.*

**3.5. Algorithm PDP-2.** In this section we consider the step-size proposed in [10, Algorithm 2], which ensures that the PDP algorithm converges in a finite number of steps. Take  $\beta > 0$  and a sequence  $\{\theta_k\} \subset \mathbb{R}_+$  such that  $\sum_j \theta_j = \infty$ , and  $\theta_k \leq \beta$  for all  $k$ . Let  $u_k$  be as Step 1(a). Consider the step-size

$$(3.5) \quad s_k \in [\eta_k, \beta_k],$$

where  $\eta_k := \theta_k / \|h(u_k)\|_1$  and  $\beta_k := \beta / \|h(u_k)\|_1$ . With this choice of  $s_k$ , we denote the PDP algorithm as PDP-2. The following result from [14, Theorem 4.5] states the convergence properties of PDP-2.

**Theorem 3.17** (Convergence of PDP-2). *Assume that  $M_P = M_D$ . Let  $\{u_k\}$  and  $\{c_k\}$  be the sequences generated by PDP-2. Suppose that the parameter sequence  $\alpha_k \geq \bar{\alpha} > 0$ . Then only one of the following cases occurs:*

- (a) *There exists a  $\bar{k}$  such that PDP-2 stops at iteration  $\bar{k}$ . As a consequence  $u_{\bar{k}}$  and  $c_{\bar{k}}$  are optimal primal and optimal dual solutions, respectively. In this situation  $\{c_k\}$  must be bounded.*
- (b) *The dual sequence  $\{c_k\}$  is unbounded. In this case,  $\{q_k\}$  converges to  $M_P$ , and  $\{u_k\}$  is bounded with all its weak accumulation points being primal solutions.*

#### 4. A CLASS OF OPTIMAL CONTROL PROBLEMS

**4.1. Problem Formulation.** In later sections, we will use the PDP algorithm to solve the optimal control of the constrained double integrator and of the free-flying

robot. These problems fall into the class of optimal control problems described in Problem (P1) below. Let the space  $\mathcal{L}^2([0, t_f]; \mathbb{R}^m)$  be as in Definition 2.7 for  $p = 2$ . Consider also the Sobolev space  $\mathcal{W}^{1,2}([0, t_f]; \mathbb{R}^n)$  as in the same definition. We consider the following class of optimal control problems.

$$(P1) \left\{ \begin{array}{l} \min \quad \frac{1}{2} \int_0^{t_f} f_0(u(t)) dt \\ \text{subject to} \quad \dot{x}(t) = f(x(t), u(t)), \\ \quad \quad \quad x(0) = x_0, \quad x(t_f) = x_f, \\ \quad \quad \quad \text{and } |u_i(t)| \leq a_i, \quad i = 1, \dots, m, \end{array} \right.$$

where the state variable  $x(t) = (x_1(t), \dots, x_n(t)) \in \mathbb{R}^n$ ,  $x \in \mathcal{W}^{1,2}([0, t_f]; \mathbb{R}^n)$ , the control variable  $u(t) = (u_1(t), \dots, u_m(t)) \in \mathbb{R}^m$ ,  $u \in \mathcal{L}^2([0, t_f]; \mathbb{R}^m)$ ,  $a_i > 0$ , for  $i = 1, \dots, m$ . Let  $f$  be linear in  $u$ , and  $f_0 : \mathbb{R}^m \rightarrow \mathbb{R}$  and  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  be  $C^1$  in their arguments. We assume that  $a_i$  is large enough so as to ensure that Problem (P1) has solutions.

**4.2. Optimality Conditions.** We now derive the first-order necessary conditions of optimality for the optimal control Problem (P1) by means of the maximum principle [26, Theorem 7.2]. The Hamiltonian function  $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$  for Problem (P1) is defined in the usual way as

$$(4.1) \quad H(x, u, \lambda, \lambda_0) := \lambda_0 f_0(u) + \lambda^T f(x, u),$$

where  $\lambda_0 \in \mathbb{R}$  and the adjoint (or costate) variable vector  $\lambda(t) := (\lambda_1(t), \dots, \lambda_n(t)) \in \mathbb{R}^n$ . We further note that  $\lambda \in \mathcal{W}^{1,2}([0, t_f]; \mathbb{R}^n)$ . In (4.1), we have dropped the dependence on  $t$  of the variables, for clarity in appearance. Also keeping up with the tradition we define

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

Next, we assume that the adjoint variable vector satisfies the differential equation

$$(4.2) \quad \dot{\lambda}(t) := -H_x[t],$$

where  $H_x = \partial H / \partial x$ . Suppose that  $(x, u) \in \mathcal{W}^{1,2}([0, t_f]; \mathbb{R}^n) \times \mathcal{L}^2([0, t_f]; \mathbb{R}^m)$  is an optimal pair for Problem (P1). Then – see [26] – there exist  $\lambda_0 \geq 0$  and a continuously differentiable adjoint variable vector  $\lambda$  as defined in (4.2), such that  $\lambda(t) \neq 0$  for all  $t \in [0, t_f]$ , and that, for all  $t \in [0, t_f]$ ,

$$(4.3) \quad u_i(t) = \underset{w \in [-a_i, a_i]}{\text{Argmin}} H(x(t), (u_1(t), \dots, w, \dots, u_m(t)), \lambda(t), \lambda_0), \quad \text{for } i = 1, \dots, m.$$

Note that in (4.3), the minimization is carried out with respect to  $w$  only, which replaces  $u_i(t)$  in the  $i$ th position of the  $u(t)$  vector. Problem (P1) is said to be *normal* when  $\lambda_0 > 0$ . When  $\lambda_0 = 0$ , the maximum principle does not convey sufficient information and Problem (P1) and its solution are referred to as *abnormal*. In [27, Example 2, pp. 2800–2801] it is shown that if the control system involving linear state ODEs is controllable then  $\lambda_0 \neq 0$ , i.e., the optimal control problem is normal. From now on, we assume that Problem (P1) is normal, and set  $\lambda_0 = 1$ , without loss of generality.

Let  $f_0(u(t)) := (u_1^2(t) + \dots + u_m^2(t))/2$ . Incorporating this special form of  $f_0$  and the linearity of  $f$  in  $u$ , (4.3) reduces to

$$(4.4) \quad u_i(t) = \begin{cases} -\sum_{j=1}^n \lambda_j(t) f_{u_i}(x(t)) & , \text{ if } -a_i < \sum_{j=1}^n \lambda_j(t) f_{u_i}(x(t)) < a_i, \\ a_i & , \text{ if } \sum_{j=1}^n \lambda_j(t) f_{u_i}(x(t)) \leq -a_i, \\ -a_i & , \text{ if } \sum_{j=1}^n \lambda_j(t) f_{u_i}(x(t)) \geq a_i, \end{cases}$$

for  $i = 1, \dots, m$ . Due to the linearity of  $f$  in  $u$ , the  $u_k$ s,  $k = 1, \dots, m$ , do not appear explicitly in  $f_{u_i} := \partial f / \partial u_i$ . For this reason, we write  $f_{u_i}(x(t))$  for  $f_{u_i}(x(t), u(t))$ . For the case when the  $i$ th control variable  $u_i(t)$  is not constrained, i.e.,  $a_i = \infty$ , the expression in (4.4) reduces to

$$u_i(t) = -\sum_{j=1}^n \lambda_j(t) f_{u_i}(x(t)).$$

**4.3. Direct Discretization and the Settings for Computations.** For computations, we discretize the problem by using the following notation. Suppose that the optimal control problem has  $n$  states and  $m$  control variables. We consider discrete approximations of the functions over the partition  $0 = t_0 < t_1 < \dots < t_N = t_f$  such that

$$t_{i+1} = t_i + \Delta t, \quad i = 0, 1, \dots, N,$$

$\Delta t := t_f/N$  and  $N$  is the number of subdivisions. Let  $u_{rj}$  be an approximation of  $u_r(t_j)$ , i.e.,  $u_{rj} \approx u_r(t_j)$ ,  $r = 1, \dots, m$ ,  $j = 0, 1, \dots, N-1$ ; similarly,  $x_{pj} \approx x_p(t_j)$ ,  $p = 1, \dots, n$ ,  $i = 0, 1, \dots, N$ .

We use the optimization modelling language AMPL [23] in coding for solving our optimal control problems and get the discrete (finite-dimensional) solution. We employ the optimization software Ipopt [30] (version 3.12.13) for solving the subproblems in the PDP algorithm, i.e. minimize the augmented Lagrangian in Step 1(a) of Algorithm 3.1. We also solve the same optimal control problems by using Ipopt on its own, in order to make comparisons with our PDP algorithm.

The AMPL–Ipopt suite was run on a Dell desktop, with the operating system Windows 10 Enterprise (version 20H2), the processor 2.40 GHz Intel Core i7 and the memory 16 GB 2666 MHz SODIMM. We have used the Ipopt options `max_iter=1000`, `tol=1e-8` and `acceptable_tol=1e-8`.

AMPL can also be paired with other optimization software, e.g., Knitro [16], SNOPT [25] and TANGO [1, 5], in solving the subproblems in the PDP algorithm.

In the next two sections, we will use the PDP algorithm to solve two optimal control problems and carry out numerical experiments with  $s_k$  chosen as in (3.4) and (3.5).

5. APPLICATION TO THE CONSTRAINED DOUBLE INTEGRATOR

In this section, we use the PDP algorithm to solve the optimal control of the constrained double integrator of a car (Problem (P2) below). We proceed to describe the classical mathematical model, and then we will present a reformulation that fits the format of Theorem 3.3. Suppose that, at time  $t$ , the position of a car modelled as a point mass travelling on a flat surface is given by  $y(t)$ . Then its velocity and acceleration are  $\dot{y}(t) = (dy/dt)(t)$ , and  $\ddot{y}(t) = (d^2y/dt^2)(t)$ , respectively. Suppose that the summation of all the external forces applied to the car is  $u(t)$ . Then by Newton's second law of motion  $\ddot{y}(t) = u(t)$  (assuming unit mass). Let  $x_1 := y$  and  $x_2 := \dot{y}$ . We impose a constraint on  $u$  that  $|u(t)| \leq a$ . We aim to minimize the squared  $\mathcal{L}^2$ -norm of the acceleration, with starting position and velocity  $x_1(0) = s_0$  and  $x_2(0) = v_0$ , and final position and velocity  $x_1(1) = s_f$  and  $x_2(1) = v_f$ , within one unit of time. This problem can then be mathematically modelled as follows.

$$(P2) \begin{cases} \min & \frac{1}{2} \int_0^1 u^2(t) dt \\ \text{subject to} & \dot{x}_1(t) = x_2(t), \quad x_1(0) = s_0, \quad x_1(1) = s_f, \\ & \dot{x}_2(t) = u(t), \quad x_2(0) = v_0, \quad x_2(1) = v_f, \quad |u(t)| \leq a, \quad \forall t \in [0, 1]. \end{cases}$$

Here the position  $x_1$  and the velocity  $x_2$  are the state variables. Assuming that we can change  $u$  the way we like, it is nothing but the control variable of the problem. Due to the box constraint on the control variable, an analytical solution for Problem (P2) is in general not possible.

**5.1. Problem (P2) Verifies (H0)–(H2).** The box constraints on  $u$  can be written as  $u \in K_1$ , where  $K_1 := \{v \in \mathcal{L}^2([0, 1], \mathbb{R}) : |v(t)| \leq a, \forall t \in [0, 1]\}$ . To use Theorem 3.3, a first step is to show that the ODE system in Problem (P2) can be equivalently written as an equality constraint of the form  $h(u) = 0$ , for a suitable function  $h$ . We do this in the following lemma.

**Lemma 5.1** (ODEs as equality constraints). *Consider the ODE system*

$$(S1) \begin{cases} \dot{x}_1(t) = x_2(t), \quad x_1(0) = s_0, \quad x_1(1) = s_f, \\ \dot{x}_2(t) = u(t), \quad x_2(0) = v_0, \quad x_2(1) = v_f. \end{cases}$$

Define

$$(5.1) \quad r_1 := v_0 + s_0 - s_f \quad \text{and} \quad r_2 := v_0 - v_f.$$

The system (S1) can be written as  $h(u) = 0$ , where  $h : \mathcal{L}^2([0, 1]; \mathbb{R}) \rightarrow \mathbb{R}^2$  is defined as

$$(5.2) \quad h(u) := \begin{bmatrix} h_1(u) \\ h_2(u) \end{bmatrix} := \begin{bmatrix} r_1 + \int_0^1 \left[ \int_0^\tau u(s) ds \right] d\tau \\ r_2 + \int_0^1 u(\tau) d\tau \end{bmatrix}.$$

*Proof.* Using the ODE constraints we re-write Problem (S1) as follows:

$$(S1') \begin{cases} x_1(1) - s_f = 0, \\ x_2(1) - v_f = 0, \end{cases}$$

where  $x_i(1)$  for  $i = 1, 2$  are defined as follows

$$(5.3) \quad \begin{aligned} x_2(t) &:= v_0 + \int_0^t u(\tau) d\tau, \\ x_1(t) &:= s_0 + \int_0^t x_2(\tau) d\tau = s_0 + \int_0^t \left[ v_0 + \int_0^\tau u(s) ds \right] d\tau. \end{aligned}$$

Note that the right hand sides in (5.3) are affine functions of  $u$ . Using this definition and (5.1), it is direct to check that  $h(u) = 0$  if and only if  $x_1(1) = s_f$  and  $x_2(1) = v_f$ .  $\square$

**Theorem 5.2** (Problem (P2) verifies (H0)–(H2)). *Let  $h$  be defined as in (5.2). Consider for Problem (P2) the dualizing parameterization  $f : \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathbb{R}^2 \rightarrow \mathbb{R}_\infty$  defined by*

$$f(u, z) := \varphi(u) + \delta_z(h(u)) + \delta_{K_1}(u).$$

where  $\varphi(u) := \frac{1}{2} \int_0^1 u^2(t) dt$ , and  $K_1 = \{u \in \mathcal{L}^2[0, 1] : |u(t)| \leq a, \forall t \in [0, 1]\}$ . Then assumptions (H0)–(H2) hold for Problem (P1).

*Proof.* It is clear that  $\varphi$  is proper and coercive. By Theorem 2.5 it is proper, coercive and w-lsc. This fact, together with the definition of  $f$ , imply that assumptions (a) and (c) from Theorem 3.3 hold. To complete the proof, we need to check that assumption (b) from Theorem 3.3 holds. Namely, we need to show that the set  $K_1 \cap h^{-1}(z)$  is w-closed. In fact, we will show that this set is w-compact, and this will provide the desired weak closedness by Fact 2.1(b). Using Theorem 2.2, it is enough to show that  $K_1 \cap h^{-1}(z) = (K_1 \cap h_1^{-1}(z_1)) \cap (K_1 \cap h_2^{-1}(z_2))$  is sequentially weakly compact. Define  $\Gamma_j := K_1 \cap h_j^{-1}(z_j)$  for  $j = 1, 2$ . We will show that each  $\Gamma_j$  is sequentially w-compact. Fix  $j \in \{1, 2\}$ . Take a sequence  $\{u_k\} \subset \Gamma_j$ . Since  $\Gamma_j \subset K_1$  and  $K_1$  is weakly compact, there exists a subsequence  $\{u_{k_l}\} \subset \{u_k\}$  s.t.  $u_{k_l} \rightharpoonup u \in K_1$ . Using (5.2) and the notation of Lemma 2.10 with  $T := 1$  and  $m := 1$ , we have that

$$(5.4) \quad h_1(u) = r_1 + \rho_1^{\varphi_1}(u, 1), \quad h_2(u) = r_1 + \eta_1^{\varphi_1}(u, 1),$$

where  $\varphi_1(s) = 1$  for every  $s \in [0, 1]$ . Because  $\{u_{k_l}\} \subset \Gamma_j$  we have that  $h_j(u_{k_l}) = z_j$ . By Lemma 2.10(c) and (5.4) we deduce that

$$z_j = \lim_{l \rightarrow \infty} h_j(u_{k_l}) = h_j(u), \quad j = 1, 2.$$

Hence,  $u \in \Gamma_j$  for  $j = 1, 2$ . This shows that both  $\Gamma_1$  and  $\Gamma_2$  are sequentially w-compact and thus the set  $K_1 \cap h^{-1}(z) = \Gamma_1 \cap \Gamma_2$  is sequentially weakly compact. By Theorem 2.2, it is w-compact and therefore w-closed. This completes the proof.  $\square$

5.2. **Numerical Solution of Problem (P2).** Using Equation (4.2), the adjoint variables for this problem can simply be written as

$$\lambda_1(t) = c_1 \quad \text{and} \quad \lambda_2(t) = -c_1 t - c_2,$$

for all  $t \in [0, 1]$ . Here  $c_1$  and  $c_2$  are real constants. Using (4.4), the optimal control for this problem is given by

$$(5.5) \quad u(t) = \begin{cases} -\lambda_2(t), & \text{if } -a \leq \lambda_2(t) \leq a, \\ a, & \text{if } \lambda_2(t) \leq -a, \\ -a, & \text{if } \lambda_2(t) \geq a, \end{cases}$$

for all  $t \in [0, 1]$ . We take  $a = 2.5$ ,  $s_0 = 0$ ,  $s_f = 0$ ,  $v_0 = 1$ , and  $v_f = 0$  in our numerical implementation. We discretize Problem (P2) over 1000 time partition points and use the PDP algorithm under both step-size of type 1 as in (3.4) and step-size of type 2 as in (3.5) to solve it.

The feasibility tolerance  $\varepsilon$  in Step 1(b) of Algorithm 3.1 is set at  $10^{-6}$ . The software package Ipopt is employed in solving the sub-problem, namely, in finding the minimizer of the Lagrangian in each iteration. We assign the parameters for the step-sizes of types 1 and 2 as follows.

- step-size of type 1:  $\alpha_k = 1$ ,  $\eta_k = 0.1$  and  $\beta_k = 1$  for all  $k$  and  $s_k$  is taken to be the midpoint of  $[\eta_k, \beta_k]$ .
- step-size of type 2:  $\alpha_k = 1$  and  $\theta_k = 1$  for all  $k$ ,  $\beta = 3$ . Using the definition of step-size of type 2 that  $\eta_k = \theta_k / \|h(u_k)\|_1$  and  $\beta_k = \beta / \|h(u_k)\|_1$ , we have  $s_k \in [\eta_k, \beta_k]$  obtained as  $s_k \in \left[ \frac{1}{\|h(u_k)\|_1}, \frac{3}{\|h(u_k)\|_1} \right]$ . Since  $c_{k+1} = c_k + (\alpha_k + 1)s_k \|h(u_k)\|_1$  by the PDP algorithm, combining the range of  $s_k$  and  $\alpha_k$ , we have that

$$c_{k+1} - c_k = (\alpha_k + 1)s_k \|h(u_k)\|_1 \in (\alpha_k + 1)[1, 3] = [2, 6].$$

That is to say, the increment of  $c_{k+1} - c_k$  is a quantity in the range of  $[2, 6]$ . In our experiments, PDP-2 uses step-size of type 2 with the parameters above, and has usually found the solution in four or five iterations. The numerical results obtained by the PDP algorithm are shown in Figure 1. One should note that the first-order optimality of the control variable in Figure 1(b) (as the necessary condition) is certified by the adjoint variable  $\lambda_2$  in Figure 1(c) via the expression in (5.5). We include the graphs of the dual function and the dual iterates by PDP-1 and PDP-2 in Figure 2.

We plot the function iterates  $u_k$  in Figure 3, where  $u_k$  are the minimizers of the dual function  $q(c_k)$  (for  $k = 0, 1, 2, 3, 4$ ).

We use different number of discretization points  $N$  to compute  $u(t)$  for Problem (P2) by our PDP algorithm. We plot the solution of  $u(t)$  obtained by the PDP algorithm with  $N = 10, 20, 100$  and  $\infty$  in Figure 4(a) and for comparison, plot the solution of  $u(t)$  by using Ipopt alone in Figure 4(b). When  $n \leq 19$ , Ipopt fails to find a solution for Problem (P2), while the PDP algorithm gives a solution when  $N$  is as small as 10.

We have also used Ipopt on its own to solve the discretization of Problem (P2) (not as a part of the PDP algorithm). These two methods achieve both 100% success

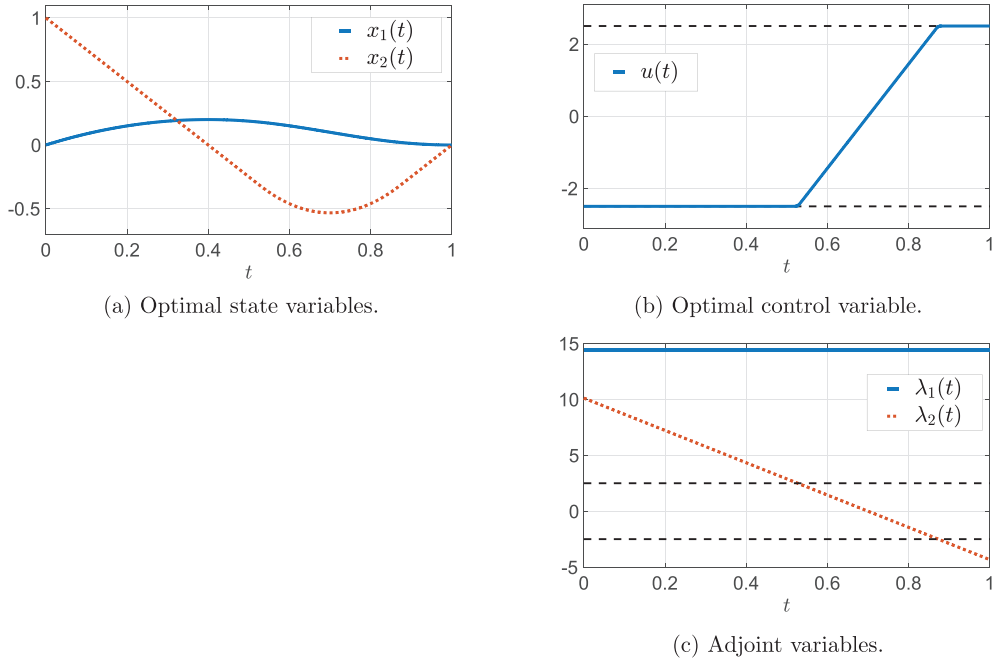


FIGURE 1. Problem ( $P2$ )—Solutions as obtained by the PDP algorithm.

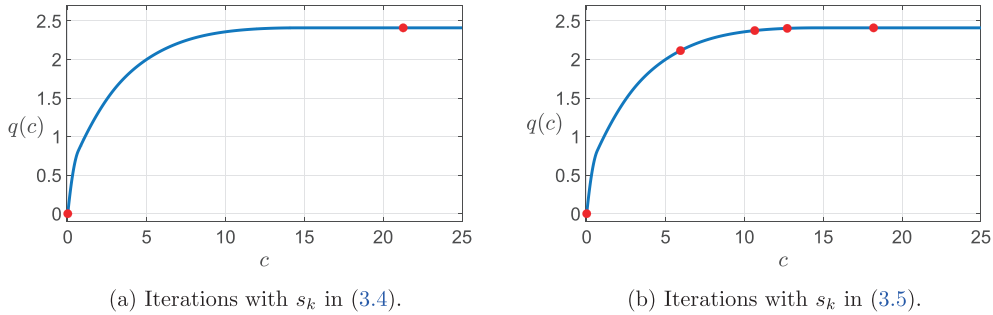


FIGURE 2. Problem ( $P2$ )—The dual function updates (shown by red dots on the blue curve representing the graph of the dual function) in each iteration of the PDP algorithm using step-sizes of type 1 and 2.

rate in finding the solution, and Ipopt alone uses less CPU time than the PDP algorithm. We note that a much more efficient method using projection techniques is provided by Bauschke, Burachik and Kaya in [4] for a class of convex optimal control problems, including Problem ( $P2$ ). Therefore, neither Ipopt nor the PDP algorithm should be the method of choice for Problem ( $P2$ ).

### 6. APPLICATION TO THE FREE FLYING ROBOT PROBLEM

The PDP algorithm can also solve non-convex problems, including the challenging Problem ( $P3$ ) below, involving the so-called free-flying robot (FFR). Problem ( $P3$ ) is highly non-convex and thus cannot be solved by existing projection methods.



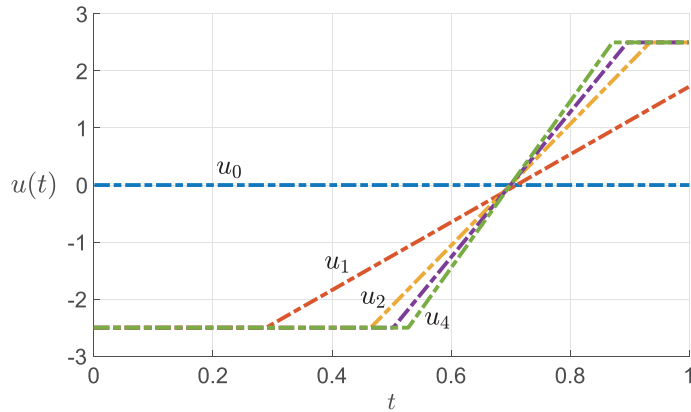


FIGURE 3. Problem (P2)—The iterations ( $u_0, u_1, u_2, u_3$  and  $u_4$ ) of  $u(\cdot)$  under the PDP algorithm with step-size of type 2. The iterate  $u_3$  is (indicated in purple but) not labelled for clarity.

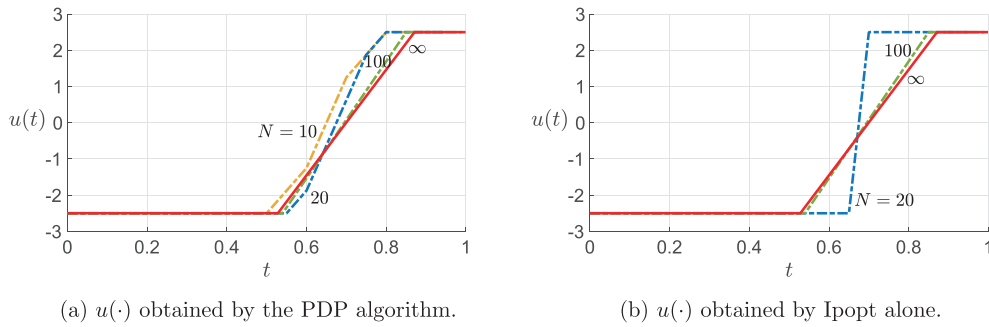


FIGURE 4. Problem (P2)—Solution  $u(\cdot)$  obtained by the PDP algorithm with step-size of type 2 and by Ipopt alone, under different number of discretization points ( $N = 10, 20$  and  $100$ ; the solution when  $N \rightarrow \infty$  is represented in red solid-line curve by the solution obtained with  $N = 1000$ ).

This warrants implementing our PDP algorithm for solving it and comparing it with the approach using Ipopt on its own.

**6.1. The Mathematical Model for Problem (P3).** The mathematical model for this problem is as follows. The aim is to minimize the fuel consumption of a robot which is moving at a constant height from an initial to a final equilibrium position. The robot can be controlled by the thrust of two jets. We use  $x_1$  and  $x_2$  for the coordinates of the FFR,  $x_4$  and  $x_5$  for the corresponding velocities,  $x_3$  for the direction of thrust,  $x_6$  for the angular velocity, and  $u_1$  and  $u_2$  for the thrusts of the two jets. The model was formulated initially in [28] and further studied in [2, 3, 29]. We use the control constraints as in [2, 3, 29]. Figures 5(a) and 5(b) respectively show a diagrammatic illustration of the model and the solution trajectory.

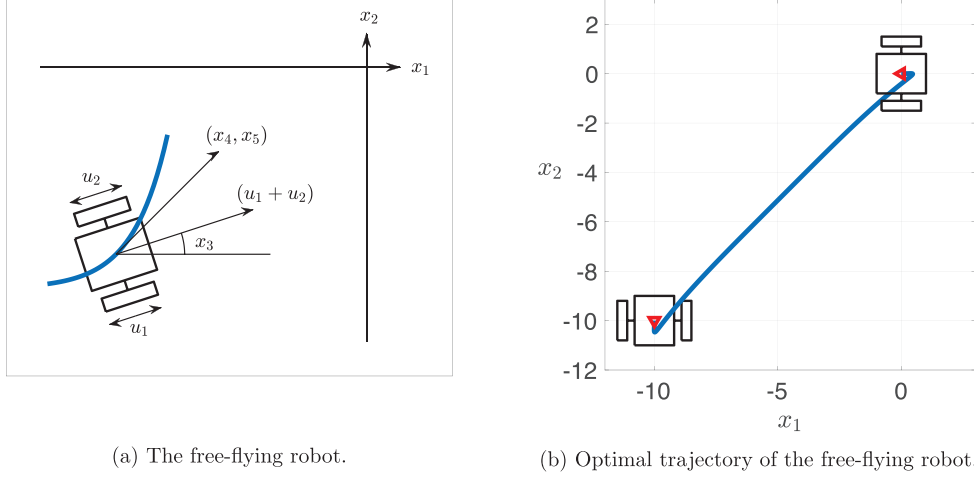


FIGURE 5. Configuration and solution trajectory of the free flying robot.

The model, as studied in [2, 3, 29], is as follows.

$$(P3) \left\{ \begin{array}{l} \min \int_0^{12} (u_1^2(t) + u_2^2(t)) dt \\ \text{s.t. } \dot{x}_1(t) = x_4(t), \\ \dot{x}_2(t) = x_5(t), \\ \dot{x}_3(t) = x_6(t), \\ \dot{x}_4(t) = (u_1(t) + u_2(t)) \cos x_3(t), \\ \dot{x}_5(t) = (u_1(t) + u_2(t)) \sin x_3(t), \\ \dot{x}_6(t) = 0.2(u_1(t) - u_2(t)), \\ x(0) = (-10, -10, \pi/2, 0, 0, 0), \quad x(12) = (0, 0, 0, 0, 0, 0), \\ |u_1(t)| \leq 0.8, \quad |u_2(t)| \leq 0.4. \end{array} \right.$$

**6.2. Formulation of the Free-flying Robot Problem.** To be able to apply the PDP algorithm for solving the FFR problem, we need to formulate (P3) in the format (P) of Theorem 3.3. With the notation of that theorem, take  $U := \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathcal{L}^2([0, 1]; \mathbb{R})$  and  $H := \mathbb{R}^6$ . The box constraints on  $u$  can be expressed using the set  $K_2 := \{u \in \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathcal{L}^2([0, 1]; \mathbb{R}) : |u_1(t)| \leq 0.8, |u_2(t)| \leq 0.4, \forall t \in [0, 1]\}$ . As in Section 5, our first step is to show that the ODE system appearing in the constraints of (P3) can be equivalently reformulated as  $h(u) = 0$  for a suitable function  $h : U \rightarrow \mathbb{R}^6$ . This fact is established in the next lemma. The idea, which is elementary albeit laborious, is to integrate the ODE system.

**Lemma 6.1** (ODEs as equality constraints). *There exists a function  $h : \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathcal{L}^2([0, 1]; \mathbb{R}) \rightarrow \mathbb{R}^6$  such that the ODE system in (P3) can be written as  $h(u) = 0$ .*

*Proof.* See the proof of Lemma 6.1 in Appendix.  $\square$

Recall our notation  $K_2 := \{(u_1, u_2) \in \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathcal{L}^2([0, 1]; \mathbb{R}) : |u_1(t)| \leq 0.8, |u_2(t)| \leq 0.4, \forall t \in [0, 1]\}$ . To verify Assumption (b) in Theorem 3.3, we need

to show that  $K_2 \cap h^{-1}(z)$  is w-closed, for  $h$  as in Lemma 6.1. We establish this in the next lemma.

**Lemma 6.2.** *Consider the ODE system and the corresponding boundary conditions given for Problem (P3) and let  $h : \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathcal{L}^2([0, 1]; \mathbb{R}) \rightarrow \mathbb{R}^6$  be as in Lemma 6.1. Then the set  $K_2 \cap h^{-1}(z)$  is w-compact and hence w-closed.*

*Proof.* See the proof of Lemma 6.2 in Appendix. □

**Corollary 6.3** (Problem (P3) verifies (H0)–(H2)). *Let  $h$  be as in Lemma 6.1. Consider for Problem (P3) the dualizing parametrization  $f : \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathcal{L}^2([0, 1]; \mathbb{R}) \times \mathbb{R}^6 \rightarrow \mathbb{R}_\infty$  defined by*

$$f(u, z) := \varphi(u) + \delta_z(h(u)) + \delta_K(u) = \varphi(u) + \delta_{h^{-1}(z) \cap K}(u).$$

*Then assumptions (H0)–(H2) hold for Problem (P3).*

*Proof.* The verification of assumptions (a) and (c) of Theorem 3.3 for (P3) is identical to the one in Theorem 5.2. Assumption (b) follows from Lemma 6.2 and the fact that  $\Gamma(0) \supset S(P) \neq \emptyset$  by [3, Section 6.2]. Indeed, the latter paper shows that there is a unique solution  $u$  of (P3). By Theorem 3.3, we conclude that (P3) verifies (H0)–(H2). □

6.2.1. *Optimality Conditions for Problem (P3).* We consider the optimality conditions for Problem (P3) as computed in Section 4.2. The Hamiltonian function  $H : \mathbb{R}^6 \times \mathbb{R}^2 \times \mathbb{R}^6 \rightarrow \mathbb{R}$  for Problem (P3) is

$$\begin{aligned} H(x, u, \lambda) &= u_1^2 + u_2^2 + \lambda_1 x_4 + \lambda_2 x_5 + \lambda_3 x_6 + \lambda_4 (u_1 + u_2) \cos x_3 \\ &\quad + \lambda_5 (u_1 + u_2) \sin x_3 + 0.2\lambda_6 (u_1 - u_2), \end{aligned}$$

where the state variable vector  $x(t) = (x_1(t), \dots, x_6(t)) \in \mathbb{R}^6$ , the control variable vector  $u(t) = (u_1(t), u_2(t)) \in \mathbb{R}^2$ . The adjoint variable vector  $\lambda(t) = (\lambda_1(t), \dots, \lambda_6(t)) \in \mathbb{R}^6$  satisfies, by Equation (4.2),

$$\begin{aligned} \lambda_1(t) &= c_1, \quad \lambda_2(t) = c_2, \quad \lambda_4(t) = -c_1 t + c_4, \quad \lambda_5(t) = -c_2 t + c_5, \quad \dot{\lambda}_6(t) = \lambda_3(t), \\ \text{and } \dot{\lambda}_3(t) &= \lambda_4(t) (u_1(t) + u_2(t)) \sin x_3(t) - \lambda_5(t) (u_1(t) + u_2(t)) \cos x_3(t), \end{aligned}$$

for all  $t \in [0, 1]$ , where  $c_1, c_2, c_4$  and  $c_5$  are real constants. By Equation (4.3), we obtain the optimal control variables as follows

$$(6.1) \quad u_1(t) = \begin{cases} -\psi_1(t) & , \text{ if } -0.8 \leq \psi_1(t) \leq 0.8, \\ 0.8 & , \text{ if } \psi_1(t) \leq -0.8, \\ -0.8 & , \text{ if } \psi_1(t) \geq 0.8, \end{cases}$$

where the *switching function* for  $u_1$  is given by

$$\psi_1(t) := \frac{1}{2} (\lambda_5(t) \cos x_3(t) + \lambda_4(t) \sin x_3(t) + 0.2\lambda_6(t));$$

$$(6.2) \quad u_2(t) = \begin{cases} -\psi_2(t) & , \text{ if } -0.4 \leq \psi_2(t) \leq 0.4, \\ 0.4 & , \text{ if } \psi_2(t) \leq -0.4, \\ -0.4 & , \text{ if } \psi_2(t) \geq 0.4, \end{cases}$$

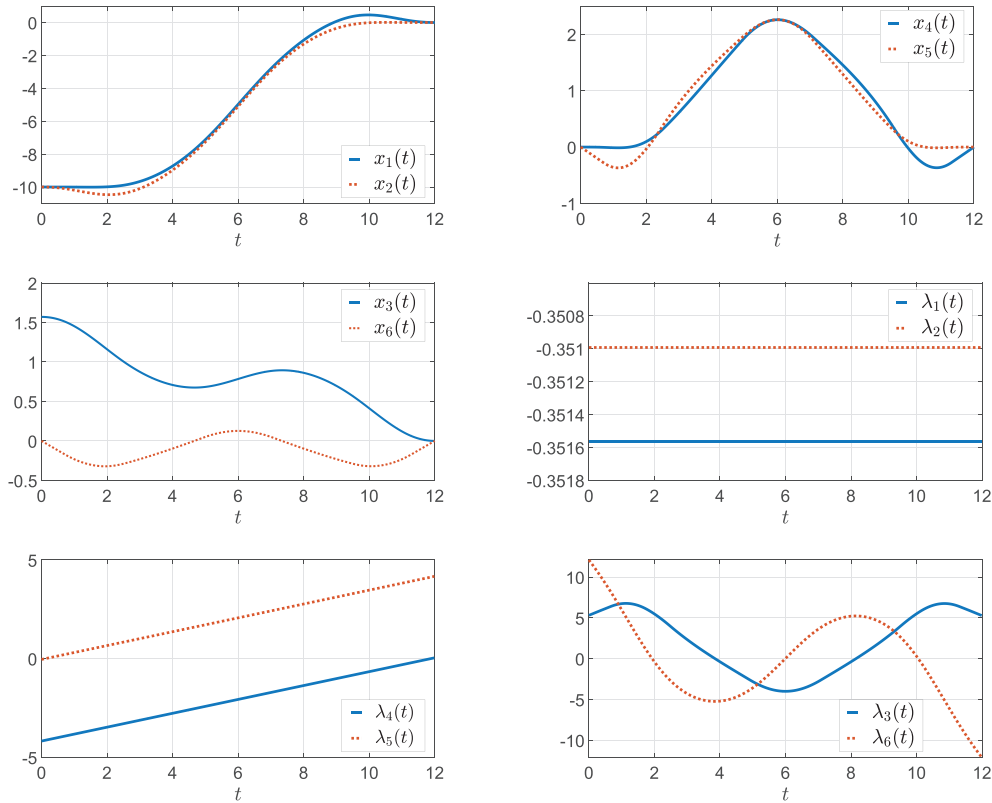


FIGURE 6. Problem (P3)—Optimal state and adjoint variables.

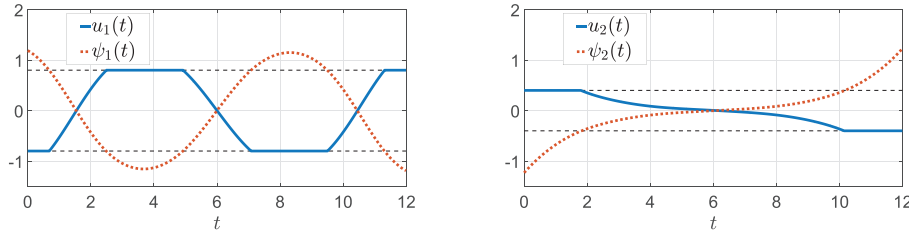
where the *switching function* for  $u_2$  is given by

$$\psi_2(t) := \frac{1}{2} (\lambda_5(t) \cos x_3(t) + \lambda_4(t) \sin x_3(t) - 0.2\lambda_6(t)).$$

**6.2.2. Numerical Solution of Problem (P3).** We discretize and solve Problem (P3) numerically as described in Section 4.3. We use the PDP algorithm with the two choices of step-sizes we have proposed, and we use Ipopt on its own to solve Problem (P3), running each of the methods 1000 times in order to get reliable statistics. We take random initial guesses generated uniformly in given intervals, such that

$$x_{pi} \in [-0.4, 0.4] \quad \text{and} \quad u_{rj} \in [-0.4, 0.4],$$

for  $p = 1, \dots, 6$ ,  $r = 1, 2$ ,  $i = 0, \dots, N$ ,  $j = 0, \dots, N - 1$ . The results for  $x(\cdot)$  and  $\lambda(\cdot)$  are shown in Figure 6, while those for  $u_i(\cdot)$  and their switching functions  $\psi_i(\cdot)$ ,  $i = 1, 2$ , are displayed in Figure 7. The graphs in Figure 7 play the role of a certificate verifying the optimality conditions given in (6.1) and (6.2). Recall that we had already included the trajectory of the free-flying robot earlier in the  $x_1x_2$ -plane in Figure 5(b).



(a) Optimal control  $u_1$  and its switching function  $\psi_1$ . (b) Optimal control  $u_2$  and its switching function  $\psi_2$ .

FIGURE 7. Problem (P3)—The optimal control variables and their switching functions.

In the PDP algorithm with the step-size in (3.4), we have taken

$$s_k = (1 + \alpha_k) \left[ \frac{1}{2} \min(\eta, \|h(u_k)\|_2) + \frac{1}{2} \max(\beta, \|h(u_k)\|_1 + \|h(u_k)\|_2) \right],$$

where  $\alpha_k = 0.4$  for all  $k$ ,  $\eta = 0.1$ , and  $\beta = 1$ , and  $h(u_{c_k})$  is the constraint function  $h$  at the current iterate  $u_{c_k}$ , for  $h$  as given in (A.12) in the proof of Lemma 6.1. In the PDP algorithm with the step-size in (3.5), we have taken  $s_k = (1 + \alpha_k)\gamma_k$ , where

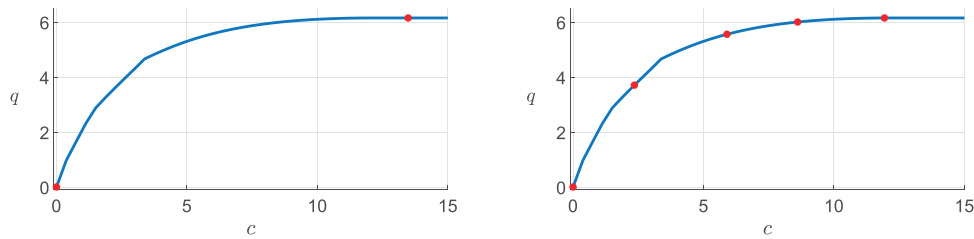
$$\gamma_k \in \left[ \frac{\theta_k}{\|h(u_k)\|_1}, \frac{\beta}{\|h(u_k)\|_1} \right],$$

and  $\alpha_k = 1$ ,  $\theta_k = 1$  for all  $k$ , and  $\beta = 2$ . The resulting dual function value iterates of the PDP algorithm, superimposed with the numerically computed graph of the dual function for Problem (P3), are displayed in Figure 8.

In Figure 9, we illustrate the iterations of  $u(t)$  using the same step-size of type 2 as in Figure 8(b). The performance of each approach, with randomly generated initial guesses, is presented in Table 1. Note that the CPU time in column 5 for Ipopt alone corresponds to the average CPU time for all runs, successful or unsuccessful. The CPU time for Ipopt alone in column 8 (last column) corresponds to the average CPU time for successful runs only.

When compared with the case of using Ipopt on its own, the PDP method achieves 100% success rate at all levels of discretization shown in Table 1. Moreover, PDP has a better performance in terms of CPU time when the number of discretization points is greater than 2000 and Ipopt is successful. When compared with the inexact restoration algorithm proposed in [3], the PDP method has a similar success rate and a better performance in terms of the CPU time, although the codes of either approach were run on different computers.

We observe that when using Ipopt alone to solve the problem, the success rate decreases as  $N$  increases. This is because in practice, the computation of a solution becomes harder as the number of optimization variables increases. When we use the PDP algorithm, the ODE system is not a hard constraint anymore, but instead it is reflected in the objective function of the minimization step. The fact that the ODE system is no longer a hard constraint seems to have a beneficial effect in terms of CPU time. More experimentation, with different types of problems, however, is needed to determine precisely whether or not this is the reason for the better performance in terms of CPU of PDP. We use different number of discretization  $N$



(a) Algorithm PDP with the step-size in (3.4). (b) Algorithm PDP with the step-size in (3.5).

FIGURE 8. Problem (P3)—The dual function updates (shown by red dots on the blue curve representing the graph of the dual function) in each iteration of Algorithm PDP using step-sizes of type 1 and 2.

TABLE 1. Problem (P3)—Success rates [%] correct to one significant figure for the PDP algorithm and Ipopt alone for the free-flying robot.

N	success rate [%]			Ave. CPU time [sec]			Ave. CPU time for Ipopt alone (when successful) [sec]
	Ipopt alone	PDP		Ipopt alone	PDP		
		$s_k$ (3.4)	$s_k$ (3.5)		$s_k$ (3.4)	$s_k$ (3.5)	
100	80	100	100	0.2	0.2	0.6	0.1
500	70	100	100	2.4	0.9	3.1	0.9
1000	50	100	100	6.0	1.6	5.8	2.2
2000	30	100	100	22.7	3.8	12.2	5.2
5000	20	100	100	108.7	9.8	30.8	17.1
10000	10	100	100	517.6	16.3	62.3	46.4

to compute  $u$  for Problem (P3) by our PDP algorithm with the same step-size of type 2 as in Figure 8(b).

We plot the solution of  $u$  with  $N = 39, 100$  and  $\infty$  (the case of  $\infty$  is represented by the solution obtained by  $N = 1000$ ) in Figure 10. The second coordinate  $u_2$  seems to be more sensitive to the number of discretization points than the first coordinate  $u_1$ . Since the curves obtained by the PDP algorithm and those obtained by Ipopt alone are indistinguishable from each other, we only show in Figure 10 those generated by the PDP algorithm. In the implementation of the PDP algorithm,  $N = 38$  does not seem to be large enough to observe the true solution pattern with the correct number of junction points in time and the more-or-less correct locations of the junctions, while Ipopt alone seems to yield solutions with correct pattern for  $N \leq 38$ . However, as discussed for Table 1, Ipopt alone fails to solve the problem over half of the time and uses more average CPU (when successful) than the PDP algorithm when  $N \geq 1000$ .

## 7. CONCLUSION AND DISCUSSION

Our work is an application of the primal–dual framework and the deflected sub-gradient algorithm studied in [14] with a numerical implementation to solve optimal control problems. Hence our framework inherits the theoretical properties of the

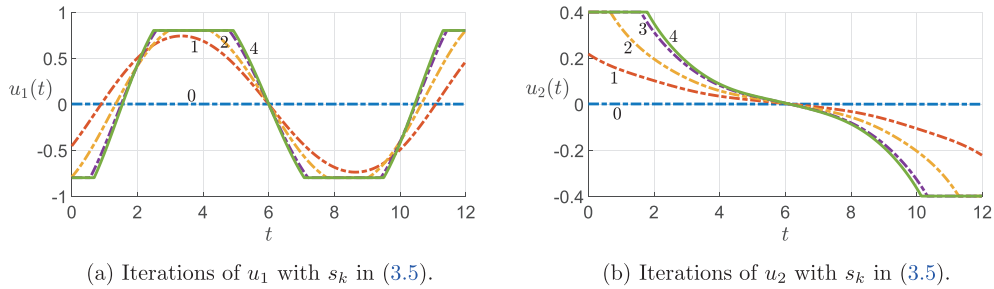


FIGURE 9. Problem (P3)—The iterations of  $u$  labelled 0–4, standing for  $u_{i,j}$ ,  $i = 1, 2$  and  $j = 0, 1, 2, 3, 4$ , obtained by algorithm PDP-2. In (a),  $u_{1,3}$  is not labelled for clarity in the appearance.

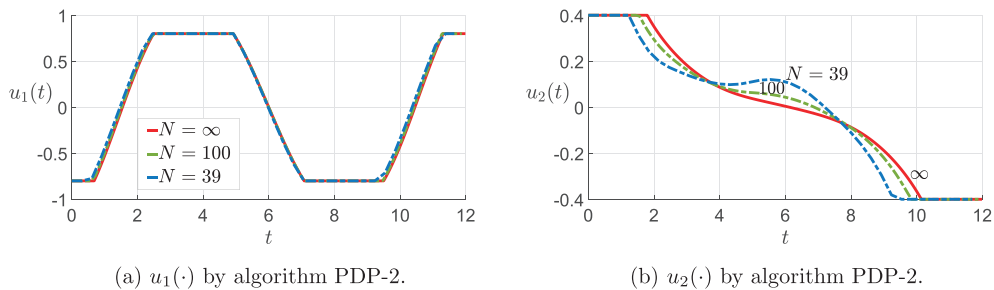


FIGURE 10. Problem (P3)—Solution  $u(\cdot)$  by algorithm PDP-2 under different number of discretization points ( $N = 39$ , and 100; the solution when  $N \rightarrow \infty$  is represented in the red solid-line curve, obtained with  $N = 1000$ ).

previous work, such as (i) strong duality (Theorem 3.9), (ii) monotone improvement of the dual function (Proposition 3.8), and (iii) every accumulation point of the primal sequence is a solution (for both step-sizes in PDP-1 and PDP-2). Moreover, PDP-2 converges in a finite number of iterations when the dual sequence is bounded (Theorem 3.17(a)).

We consider infinite dimensional optimization problems which satisfy the assumptions (H0)–(H2). We have presented here a systematic technique to verify these assumptions in the framework of very general types of optimization problems (Theorem 3.3). We show this for problems (P2) and (P3) (see Theorem 5.2 and Corollary 6.3). Particularly, we have demonstrated how to reformulate the ODE constraints in Problems (P2) and (P3) as equality constraints so that the assumptions (H0)–(H2) can be verified.

Problem (P2) is the constrained optimal control of the double integrator and Problem (P3) is the more challenging free-flying robot, again constrained. We illustrate the iterations of the control variables using our PDP algorithm, using Ipopt in solving its subproblems, for Problem (P2) in Figure 3 and for Problem (P3) in Figure 9. Compared with using Ipopt alone, our PDP algorithm shows a better performance in solving the challenging flying robot problem in terms of the CPU time and in the case of increased number of discretization points (see Table 1).

In PDP-1, the step-size  $s_k$  as in (3.4) gets smaller in each iteration and the increments in the penalty parameter  $c_k$  gets smaller accordingly. In numerical practice, PDP-1 has a much bigger increment in  $c_k$  in the first one or two iterates. Hence, PDP-1 can often find the optimal dual solution in just a few iterates. Although PDP-1 approaches the dual solution very quickly in the first few iterates, it becomes sluggish in the following iterates since  $c_k$  is incremented very slowly.

Algorithm PDP-2 uses the step-size  $s_k$  as in (3.5). The increase in the penalty parameter  $c_k$  is small in the initial iterates, compared to PDP-1. PDP-2 resembles the penalty method with constant increments in  $c_k$ . In practice, PDP-2 approaches the dual optimal solution slowly but steadily. To avoid the slow progress of PDP-1 in later iterations, we would suggest a hybrid strategy, which applies PDP-1 in the first few iterates and then switches to PDP-2.

Apart from the examples in this paper, our algorithm should be applicable to more general optimal control problems, for example the more challenging problems involving mixed state-control constraints or even pure state constraints.

The PDP algorithm could be used as a theoretical tool that provides dual information on a given problem. When PDP is applied to solve problems where the analytical solution can be found relatively easily (e.g., the unconstrained double integrator problem), then the use of the algorithm is likely to reveal new properties of the problem, especially those arising from duality.

Practically speaking, because we are doing the iterations with discretized functions, there is already a level of inexactness introduced into the subproblem solutions. The numerical experiments show that this kind of inexactness is dealt with successfully. Other types of inexact versions could be done by extending the work presented in [12], which deals with finite dimensional problems. The cases we consider in the present paper are infinite dimensional and hence more challenging. Theoretical investigation of inexactness in infinite dimensions remains an open problem and hence the topic of future research.

## APPENDIX Appendix A

### Proof of Lemma 2.10

*Proof.* Assume that  $\{u_k\} \subset (\mathcal{L}^2([0, T]; \mathbb{R}))^m$  is such that  $u_k \rightharpoonup u$ . Then for all  $j = 1, \dots, m$  we have  $u_k^j \rightharpoonup u^j$  (weakly in  $\mathcal{L}^2([0, T]; \mathbb{R})$ ). For  $t \in [0, T]$ , define  $\xi_t(s) = 0$  for  $s \in (t, T]$  and  $\xi_t(s) = 1$  for  $s \in [0, t]$ . Then  $\xi_t \in \mathcal{L}^2([0, T]; \mathbb{R})$  and it is easy to check that  $\|\xi_t\|_2 = \sqrt{t}$ . The weak convergence yields

$$(A.1) \quad \lim_{k \rightarrow \infty} \int_0^t u_k^j(s) ds = \lim_{k \rightarrow \infty} \int_0^T u_k^j(s) \xi_t(s) ds = \int_0^T u^j(s) \xi_t(s) ds = \int_0^t u^j(s) ds,$$

where we used the definition of  $\xi_t$  in the first and last equality, and the assumption of weak convergence in the second one. For each  $j = 1, \dots, m$ , define  $f_k^j, f^j : [0, T] \rightarrow \mathbb{R}$  as  $f_k^j(t) := \int_0^t u_k^j(s) ds$  and  $f^j(t) := \int_0^t u^j(s) ds$ . By (A.1),

$$(A.2) \quad \lim_{k \rightarrow \infty} f_k^j(t) = f^j(t),$$

for every  $t \in [0, T]$  and each  $j = 1, \dots, m$ . We will apply Theorem 2.9 to the sequence  $\{f_k^j\}$  for each  $j = 1, \dots, m$ . Since  $\{u_k\}$  converges weakly, the set  $U_0 :=$



$\{u_k : k \in \mathbb{N}\} \cup \{u\}$  is weakly compact, and hence bounded by Corollary 2.4. Therefore, there exists  $L_0 > 0$  such that  $\|u_k^j\|_2 \leq L_0$  for all  $k \in \mathbb{N}$ ,  $j = 1, \dots, m$ . We show next that the sequence  $\{f_k^j\}$  is bounded over  $[0, T]$  for each  $j = 1, \dots, m$ . Indeed, for every  $t \in [0, T]$  we can use Cauchy-Schwartz to write

$$(A.3) \quad |f_k^j(t)| = \left| \int_0^t u_k^j(s) ds \right| = |\langle u_k^j, \xi_t \rangle_{\mathcal{L}^2([0, T]; \mathbb{R})}| \leq \|u_k^j\|_2 \|\xi_t\|_2 \leq L_0 \sqrt{t} \leq L_0 \sqrt{T},$$

where  $\langle \cdot, \cdot \rangle_{\mathcal{L}^2([0, T]; \mathbb{R})}$  denotes the scalar product in  $\mathcal{L}^2([0, T]; \mathbb{R})$ . This establishes the desired boundedness. Using now Theorem 2.9 we deduce that

$$(A.4) \quad \int_0^t f^j(s) ds = \lim_{k \rightarrow \infty} \int_0^t f_k^j(s) ds,$$

which by definition of  $f_k^j, f^j$  re-writes as

$$(A.5) \quad \int_0^t \int_0^s w^j(r) dr ds = \lim_{k \rightarrow \infty} \int_0^t \int_0^s u_k^j(r) dr ds,$$

Using the definition of  $\pi(u)$  and (A.5) we deduce that

$$\lim_{k \rightarrow \infty} \pi(u_k)(t) = \pi(u_k)(t),$$

for every  $t \in [0, T]$ . Since  $\varphi$  is continuous, we further have

$$(A.6) \quad \lim_{k \rightarrow \infty} \varphi(\pi(u_k)(t)) = \varphi(\pi(u_k)(t)),$$

for every  $t \in [0, T]$ . Consider the functions  $\omega_k := \varphi(\pi(u_k))$  and  $\omega := \varphi(\pi(u))$ . By definition,  $\omega_k, \omega : [0, T] \rightarrow \mathbb{R}$ . We claim that  $\omega_k, \omega \in \mathcal{L}^2([0, T]; \mathbb{R})$ . Indeed, the boundedness assumption on  $\varphi$  gives

$$\|\omega_k\|_2^2 = \int_0^T |\varphi(\pi(u_k)(s))|^2 ds \leq L_1^2 T.$$

An identical argument shows that  $\omega \in \mathcal{L}^2([0, T]; \mathbb{R})$ . Note that this claim implies that  $\omega_k, \omega \in \mathcal{L}^2([0, t]; \mathbb{R})$  for every  $t \in [0, T]$ . Now we claim that  $\omega_k \rightarrow \omega$  strongly in  $\mathcal{L}^2([0, t]; \mathbb{R})$  for every  $t \in [0, T]$ . Namely, we claim that

$$(A.7) \quad \lim_{k \rightarrow \infty} \|\omega_k - \omega\|_{\mathcal{L}^2([0, t]; \mathbb{R})}^2 = \lim_{k \rightarrow \infty} \int_0^t |\omega_k(s) - \omega(s)|^2 ds = 0,$$

for every  $t \in [0, T]$ . Indeed, for  $s \in [0, T]$  define  $\Delta_k(s) := |\omega_k(s) - \omega(s)|^2$ . The definition of the functions  $\omega_k, \omega$  and (A.6) imply that  $\lim_{k \rightarrow \infty} \Delta_k(s) = 0$  for every  $s \in [0, T]$ . Using the boundedness of  $\varphi$  and the definitions, we also have that

$$\Delta_k(s) = |\omega_k(s)|^2 + |\omega(s)|^2 + 2|\omega_k(s)||\omega(s)| \leq 4L_1^2,$$

for every  $s \in [0, T]$ . Now we can apply Theorem 2.9(b), (A.6), and the definitions to deduce that

$$\lim_{k \rightarrow \infty} \int_0^t \Delta_k(s) ds = \lim_{k \rightarrow \infty} \int_0^t |\omega_k(s) - \omega(s)|^2 ds = \int_0^t \lim_{k \rightarrow \infty} \Delta_k(s) ds = 0,$$

for every  $t \in [0, T]$ . This establishes (A.7). Using the definition of  $\xi_t$ , the above expression re-arranges as follows.

$$\begin{aligned} 0 &= \lim_{k \rightarrow \infty} \int_0^t |\omega_k(s) - \omega(s)|^2 ds = \int_0^T \xi_t(s) |\omega_k(s) - \omega(s)|^2 ds, \\ &= \int_0^T (\xi_t(s))^2 |\omega_k(s) - \omega(s)|^2 ds = \int_0^T |\xi_t(s)\omega_k(s) - \xi_t(s)\omega(s)|^2 ds, \end{aligned}$$

showing that the sequence  $\{\xi_t \omega_k\}$  converges strongly in  $\mathcal{L}^2([0, T]; \mathbb{R})$  to  $\xi_t \omega$ . To complete the proof, we will use Proposition 2.8, for the space  $X := \mathcal{L}^2([0, T]; \mathbb{R})$ , the strongly convergent sequence  $\{\xi_t \omega_k\}$ , and each of the weakly convergent sequences  $\{u_k^j\}$  for  $j = 1, \dots, m$ . This proposition implies that

$$\begin{aligned} \lim_{k \rightarrow \infty} \int_0^t \omega_k(s) u_k^j(s) ds &= \lim_{k \rightarrow \infty} \int_0^T \xi_t(s) \omega_k(s) u_k^j(s) ds \\ \text{(A.8)} \qquad \qquad \qquad &= \int_0^T \xi_t(s) \omega(s) u^j(s) ds = \int_0^t \omega(s) u^j(s) ds. \end{aligned}$$

the above expression and the definition of  $\omega_k, \omega$  imply that (2.1) holds. Finally (2.2) will follow from applying Theorem 2.9(b) to the sequence  $\{\eta_j^\varphi(u_k, \cdot)\}$ . Indeed, (A.8) means that  $\lim_{k \rightarrow \infty} \eta_j^\varphi(u_k, s) = \eta_j^\varphi(u, s)$  for every  $s \in [0, T]$ . We now use an argument similar to the one in (A.3) to show the boundedness of the sequence over  $[0, T]$ . Indeed, fix  $t \in [0, T]$ .

$$\begin{aligned} |\eta_j^\varphi(u_k, t)| &= \left| \int^t \omega_k(s) u_k^j(s) ds \right| = \left| \int^T \xi_t(s) \omega_k(s) u_k^j(s) ds \right| \\ &= |\langle u_k^j, \xi_t \omega_k \rangle_{\mathcal{L}^2([0, T]; \mathbb{R})}| \leq \|u_k^j\|_2 \|\xi_t \omega_k\|_2 \leq L_0 L_1 \sqrt{t} \leq L_0 L_1 \sqrt{T}, \end{aligned}$$

Now Theorem 2.9(b) yields

$$\lim_{k \rightarrow \infty} \int_0^t \eta_j^\varphi(u_k, s) ds = \int_0^t \eta_j^\varphi(u, s) ds,$$

which is (2.2). The proof is complete. □

**Proof of Lemma 6.1**

*Proof.* Fix  $T := 12$ . The desired function  $h$  will be obtained by repeatedly applying the Fundamental Theorem of Calculus to each equation of the ODE system in Problem (P3), starting with the last equation. Consider the last ODE equation in (P3), together with its boundary conditions, namely

$$\text{(A.9)} \quad \begin{cases} \dot{x}_6(t) = 0.2(u_1(t) - u_2(t)) =: g_6(u(t)), \\ x_6(0) = 0, \quad x_6(T) = 0, \end{cases}$$

By Fundamental Theorem of Calculus this system is equivalent to  $G_6(u(T)) = 0$ , where  $G_6(u(t)) := \int_0^t g_6(u(s)) ds$ . Hence, for every  $u \in \mathcal{L}^2([0, T]; \mathbb{R}) \times \mathcal{L}^2([0, T]; \mathbb{R})$  we define

$$h_6(u) := G_6(u(T)) = \int_0^T g_6(u(s)) ds = a_1 \int_0^T u_1(s) ds - a_2 \int_0^T u_2(s) ds,$$

where we used the definition of  $g_6$  in the third equality with the notation  $a_1 := a_2 := 0.2$ . Thus,  $h_6 : \mathcal{L}^2([0, T]; \mathbb{R}) \times \mathcal{L}^2([0, T]; \mathbb{R}) \rightarrow \mathbb{R}$ . In particular, we can write the solution  $x_6(\cdot)$  of system (A.9) as a function of  $u$ . Namely,

$$(A.10) \quad x_6(t) = \int_0^t g_6(u(s)) ds = a_1 \int_0^t u_1(s) ds - a_2 \int_0^t u_2(s) ds.$$

Since  $x_6$  is a function of  $u$ , a similar procedure can be used for the third equation in (P3) and its boundary conditions. Indeed, using (A.10)

$$(A.11) \quad \text{the system } \left\{ \begin{array}{l} \dot{x}_3(t) = x_6(t) = \int_0^t g_6(u(s)) ds, \\ x_3(0) = \pi/2, x_3(T) = 0, \end{array} \right\}, \text{ is equivalent to } G_3(u(T)) = 0,$$

where  $G_3(u(t)) := \pi/2 + \int_0^t \int_0^r g_6(u(s)) ds dr$  and  $g_6$  is defined in the previous system. We can thus write, for every  $u \in \mathcal{L}^2([0, T]; \mathbb{R}) \times \mathcal{L}^2([0, T]; \mathbb{R})$

$$\begin{aligned} h_3(u) := G_3(u(T)) &= \pi/2 + \int_0^T \int_0^r g_6(u(s)) ds dr \\ &= \pi/2 + a_1 \int_0^T \int_0^r u_1(s) ds dr - a_2 \int_0^T \int_0^r u_2(s) ds dr. \end{aligned}$$

Hence,  $h_3 : \mathcal{L}^2([0, T]; \mathbb{R}) \times \mathcal{L}^2([0, T]; \mathbb{R}) \rightarrow \mathbb{R}$ . We proceed now to define the remaining  $h_i$ 's. As a consequence of our last construction, we have that we can write the solution  $x_3(\cdot)$  of system (A.11) as a function of  $u$ . Namely,  $x_3(t) = G_3(u(t))$ . Therefore, we have the following equivalence

$$\left\{ \begin{array}{l} \dot{x}_5(t) = (u_1(t) + u_2(t)) \sin x_3(t) \\ = (u_1(t) + u_2(t)) \sin G_3(u(t)) =: g_5(u(t)), \\ x_5(0) = 0, x_5(T) = 0, \end{array} \right\}, \text{ if and only if } G_5(u(T)) = 0,$$

where  $G_5(u(t)) := \int_0^t g_5(u(s)) ds$ . As above, use the definition of  $g_5$  to define

$$\begin{aligned} h_5(u) := G_5(u(T)) &= \int_0^T g_5(u(s)) ds \\ &= \int_0^T u_1(s) \sin G_3(u(s)) ds + \int_0^T u_2(s) \sin G_3(u(s)) ds. \end{aligned}$$

Using again the fact that  $x_3(t) = G_3(u(t))$ , write

$$\left\{ \begin{array}{l} \dot{x}_4(t) = (u_1(t) + u_2(t)) \cos x_3(t) \\ = (u_1(t) + u_2(t)) \cos G_3(u(t)) =: g_4(u(t)), \\ x_4(0) = 0, x_4(T) = 0, \end{array} \right\}, \text{ is equivalent to } G_4(u(T)) = 0,$$

where  $G_4(u(t)) := \int_0^t g_4(u(s))ds$ . So we define

$$h_4(u) := G_4(u(T)) := \int_0^T g_4(u(s))ds = \int_0^T (u_1(t) + u_2(t)) \cos G_3(u(t))dt.$$

Again, note that the system

$$\left\{ \begin{array}{l} \dot{x}_2(t) = x_5(t) = \int_0^t g_5(u(s))ds =: g_2(u(t)), \\ x_2(0) = -10, x_2(T) = 0, \end{array} \right\}, \text{ is equivalent to } G_2(u(T)) = 0,$$

where  $G_2(u(t)) := -10 + \int_0^t \int_0^r g_5(u(s))dsdr$ . So we define

$$h_2(u) := G_2(u(T)) := -10 + \int_0^T g_2(u(s))ds.$$

Finally, we can write

$$\left\{ \begin{array}{l} \dot{x}_1(t) = x_4(t) = \int_0^t g_4(u(s))ds =: g_1(u(t)), \\ x_1(0) = -10, x_1(T) = 0, \end{array} \right\}, \text{ is equivalent to } G_1(u(T)) = 0,$$

where  $G_1(u(t)) := -10 + \int_0^t \int_0^r g_4(u(s))dsdr$ . So we define

$h_1(u) := -10 + \int_0^T g_1(u(s))ds$ . Altogether, the ODE system in (P3) can be rewritten in terms of  $u$  as

$$(A.12) \quad h(u) = (G_1(u(T)), \dots, G_6(u(T))) = (h_1(u), \dots, h_6(u)) = 0 \in \mathbb{R}^6.$$

□

**Proof of Lemma 6.2**

*Proof.* The w-closedness will follow from the w-compactness and Fact 2.1(b). We proceed to establish the w-compactness. For  $h$  as in Lemma 6.1, write

$$\Gamma(z) := \{u \in K_2 : h(u) = z\} = h^{-1}(z) \cap K_2 = \bigcap_{j=1}^6 [h_j^{-1}(z_j) \cap K_2].$$

Call  $\Gamma_j := h_j^{-1}(z_j) \cap K_2$  for  $j = 1, \dots, 6$ . We will show that each  $\Gamma_j$  is weakly compact for  $j = 1, \dots, 6$ . Indeed, with the notation of Lemma 2.10, and the definition of  $h_j$  given in Lemma 6.1, we have that

$$(A.13) \quad \begin{aligned} h_1(u) &= -10 + \rho_1^{\varphi^c}(u, T) + \rho_2^{\varphi^c}(u, T), & h_2(u) &= -10 + \rho_1^{\varphi^s}(u, T) + \rho_2^{\varphi^s}(u, T), \\ h_3(u) &= \pi/2 + \rho_1^{\varphi^1}(u, T) - \rho_2^{\varphi^1}(u, T), & h_4(u) &= \eta_1^{\varphi^c}(u, T) + \eta_2^{\varphi^c}(u, T), \\ h_5(u) &= \eta_1^{\varphi^s}(u, T) + \eta_2^{\varphi^s}(u, T), & h_6(u) &= 0.2(\eta_1^{\varphi^1}(u, T) - \eta_2^{\varphi^1}(u, T)), \end{aligned}$$

where  $\varphi_c := \cos(\cdot)$ ,  $\varphi_s := \sin(\cdot)$  and  $\varphi_1(r) = 1$  for every  $r \in [0, T]$ . Fix  $j \in \{1, \dots, m\}$ . By Theorem 2.2,  $\Gamma_j$  is weakly compact if and only if it is sequentially weakly compact. Namely, if and only if, for every sequence  $\{u_k\} \subset \Gamma_j$ , there exists a subsequence  $\{u_{k_l}\} \subset \{u_k\}$  s.t.  $u_{k_l} \rightharpoonup u \in \Gamma_j$ . Take now any sequence  $\{u_k\} \subset \Gamma_j$ . Since  $\Gamma_j \subset K_2$  and  $K_2$  is weakly compact, there exists a subsequence  $\{u_{k_l}\} \subset \{u_k\}$  s.t.  $u_{k_l} \rightharpoonup u \in K_2$ . Because  $\{u_{k_l}\} \subset \Gamma_j$  we have that  $h_j(u_{k_l}) = z_j$ . By (2.1) and (2.2) in Lemma 2.10 we deduce from (A.13) that

$$z_j = \lim_{l \rightarrow \infty} h_j(u_{k_l}) = h_j(u),$$

for  $j = 1, \dots, 6$ . Hence,  $u \in \Gamma_j$  as wanted and  $\Gamma_j$  is weakly compact for all  $j = 1, \dots, m$ . Therefore  $\Gamma(z)$  is w-compact and hence w-closed.  $\square$

#### ACKNOWLEDGMENTS

The authors offer their warm thanks to the Editor for their efficient handling of the paper. They are also indebted to an anonymous reviewer whose comments improved the manuscript. Xuemei Liu was supported by an Australian Government Research Training Program Scholarship.

#### REFERENCES

- [1] R. Andreani, E. G. Birgin, J. M. Martínez and M. L. Schuverdt, *On augmented Lagrangian methods with general lower-level constraints*, SIAM J. Optim. **18** (2007), 1286–1309.
- [2] N. Banihashemi and C. Y. Kaya, *Inexact restoration for Euler discretization of box-constrained optimal control problems*, J. Optim. Theory Appl. **156** (2013), 726–760.
- [3] N. Banihashemi and C. Y. Kaya, *Inexact Restoration and adaptive mesh refinement for optimal control*, J. Indust. Man. Optim. **10** (2014), 521–542.
- [4] 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, pp. 45–68.
- [5] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*, SIAM Publications, 2014.
- [6] H. Brezis, *Functional Analysis, Sobolev Spaces and Partial Differential Equations*, Springer, Berlin, 2011.
- [7] R. S. Burachik, W. P. Freire and C. Y. Kaya, *Interior epigraph directions method for nonsmooth and nonconvex optimization via generalized augmented Lagrangian duality*, J. Global Optim. **60** (2014), 501–529.
- [8] R. S. Burachik, R. N. Gasimov, N. A. Ismayilova and C. Y. Kaya, *On a modified subgradient algorithm for dual problems via sharp augmented Lagrangian*, J. Global Optim. **34** (2006), 55–78.
- [9] R. S. Burachik, A. N. Iusem and J. G. Melo, *A primal dual modified subgradient algorithm with sharp Lagrangian*, J. Global Optim. **46** (2010), 55–78.
- [10] R. S. Burachik, A. N. Iusem and J. G. Melo, *An inexact modified subgradient algorithm for primal-dual problems via Augmented Lagrangians*, J. Optim. Theory Appl. **157** (2013), 108–131.
- [11] R. S. Burachik and C. Y. Kaya, *A deflected subgradient method using a general augmented Lagrangian duality with implications on penalty methods*, in: Variational Analysis and Generalized Differentiation in Optimization and Control, Springer Optimization and Its Applications, R.S. Burachik, J. C. Yao (eds.), vol. 47, Springer, New York, 2010, pp. 109–132.
- [12] R. S. Burachik, C. Y. Kaya and M. Mammadov, *An inexact modified subgradient algorithm for nonconvex optimization*, Comput. Optim. Appl. **45** (2010), 1–24.

- [13] R. S. Burachik, C. Y. Kaya and C. J. Price, *A primal–dual penalty method via rounded weighted- $\ell_1$  Lagrangian duality*, Optimization **71** (2022), 3981–4017.
- [14] R. S. Burachik and X. Liu, *An inexact deflected subgradient algorithm in infinite dimensional spaces*, arXiv, <https://arxiv.org/abs/2302.02072>, 2023.
- [15] R. S. Burachik and A. M. Rubinov, *Abstract convexity and augmented Lagrangians*, SIAM J. Optim. **18** (2007), 413–436.
- [16] R. H. Byrd, J. Nocedal and R. A. Waltz, *KNITRO: An integrated package for nonlinear optimization*, in: Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma (ed.), Springer, 2006, pp. 35–59.
- [17] V. F. Demyanov, F. Giannessi and V. V. Karelin, *Optimal control problems via exact penalty functions*, J. Glob. Optim. **12** (1998), 215–223.
- [18] V. F. Demyanov, F. Giannessi and V. Karelin, *On the penalization approach to optimal control problems*, IFAC Proc. **33** (2000), 71–74.
- [19] V. F. Demyanov, F. Giannessi and V. V. Karelin, *Optimal control problems and penalization*, in: Nonlinear Optimization and Related Topics, G. D. Pillo and F. Giannessi (eds), Springer, Boston, 2000. pp. 67–78.
- [20] V. F. Demyanov and G. S. Tamasyan, *Exact penalty functions in isoperimetric problems*, Optimization **60** (2011), 153–177.
- [21] M. V. Dolgopolik, *Augmented Lagrangian functions for cone constrained optimization: the existence of global saddle points and exact penalty property*, J. Glob. Optim. **71** (2018), 237–296.
- [22] M. V. Dolgopolik, *A unified approach to the global exactness of penalty and augmented Lagrangian functions I: parametric exactness*, J. Optim. Theory Appl. **176** (2018), 728–744.
- [23] R. Fourer, D. M. Gay and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming, Second Edition*. Brooks/Cole Publishing Company / Cengage Learning, 2003.
- [24] R. N. Gasimov, *Augmented Lagrangian duality and nondifferentiable, optimization methods in nonconvex programming*, J. Global Optim. **24** (2002), 187–203.
- [25] P. E. Gill, W. Murray and M. A. Saunders, *SNOPT: an SQP algorithm for large-scale constrained optimization*, SIAM Rev. **47** (2005), 99–131.
- [26] M. R. Hestenes, *Calculus of Variations and Optimal Control Theory*, John Wiley & Sons, New York, 1966.
- [27] C. Y. Kaya and J. L. Noakes, *Leapfrog for optimal control*, SIAM J. Numer. Anal. **46** (2008), 2795–2817.
- [28] Y. Sakawa, *Trajectory planning of a free-flying robot by using the optimal control*, Optim. Control Appl. Methods **20** (1999), 235–248.
- [29] G. Vossen and H. Maurer, *On  $L^1$ -minimization in optimal control and applications to robotics*, Optim. Control Appl. Methods **27** (2006), 301–321.
- [30] 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.

R. S. BURACHIK

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

*E-mail address:* `regina.burachik@unisa.edu.au`

C. Y. KAYA

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

*E-mail address:* `yalcin.kaya@unisa.edu.au`

X. LIU

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

*E-mail address:* `xuemei.liu@mymail.unisa.edu.au`