DIRECT METHOD TO SOLVE LINEAR-QUADRATIC OPTIMAL CONTROL PROBLEMS

MOHAMED ALIANE° AND MOHAND BENTOBACHE
Laboratory of Pure and Applied Mathematics
University of Laghouat, Bp 37G
Ghardaia road, 03000, Laghouat, Algeria

NACIMA MOUSSOUNI
Laboratory L2CSP, University of Tizi-Ouzou, 15000, Tizi-Ouzou, Algeria

PHILIPPE MARTHON
IRIT-ENSEEIHT, Université Fédérale Toulouse-Midi-Pyrénées, France

(Communicated by Chongyang Liu)

Abstract. In this work, we have proposed a new approach for solving the linear-quadratic optimal control problem, where the quality criterion is a quadratic function, which can be convex or non-convex. In this approach, we transform the continuous optimal control problem into a quadratic optimization problem using the Cauchy discretization technique, then we solve it with the active-set method. In order to study the efficiency and the accuracy of the proposed approach, we developed an implementation with MATLAB, and we performed numerical experiments on several convex and non-convex linear-quadratic optimal control problems. The obtained simulation results show that our method is more accurate and more efficient than the method using the classical Euler discretization technique. Furthermore, it was shown that our method fastly converges to the optimal control of the continuous problem found analytically using the Pontryagin’s maximum principle.

1. Introduction. The theory of optimal control is a branch of mathematics which consists in finding the best control which guides a given system, such as a car, a space shuttle, or a chemical reaction, on which we have an action, from an initial state to a final one. The optimal control theory can be applied in various fields: aeronautics and aerospace [1, 2, 3, 6, 12, 18], mechanics [8, 10], agriculture [13, 14], etc.

There are two types of numerical methods in optimal control, indirect methods and direct ones. Indirect methods consist in solving numerically the considered problem by the shooting method: a boundary value problem is first obtained by the application of the Pontryagin’s maximum principle [16], then a non-linear system of equations is solved for finding an approximate solution. Direct methods are based on partial or full discretization of the time interval, they transform the continuous optimal control problem to a linear or non-linear optimization problem [15].
In [2], the Cauchy discretization technique is combined with an interior-point method to efficiently solve the linear optimal control problem modelling a rectilinear motion of a rocket and in [19], the Cauchy discretization technique is combined with the hybrid direction algorithm developed in [5] to solve the linear optimal control problem.

In [11], the authors proposed a primal-dual method for solving the linear-quadratic optimal control problem with terminal linear constraints, where the quality criterion to be minimized is a convex quadratic function. In [7], the authors proposed a method based on approximation sets for solving the optimal control problem, where the quality criterion to be minimized is a differentiable concave function.

In this work, we propose a new algorithm called “Cauchy Discretization Algorithm” (CDA) for solving Linear-Quadratic Optimal Control (LQOC) problems, where the quality criterion to be maximized is a quadratic function which can be convex or non-convex. The principle of this algorithm is as follows: first a change of variable is made using the Cauchy formula for solving systems of linear differential equations with initial conditions; then a discretization phase is applied and a quadratic programming problem is obtained. Finally, the quadratic program is solved using the active-set method.

This article is structured as follows: In Section 2, we present the linear-quadratic optimal control problem. In Section 3, we present the proposed algorithm for solving LQOC problems. In Section 4, we illustrate our approach with four convex and non-convex LQOC problems and we present numerical experiments which compare our method with the analytical method (AM) and the classical Euler Discretization Method (EDM). Finally, we conclude this paper and give some future works.

2. Statement of the problem. Let \( n \) and \( m \) be two nonnegative integers, \( A \in \mathcal{M}_n(\mathbb{R}) \), \( B \in \mathcal{M}_{n,m}(\mathbb{R}) \) and \( x_0 \in \mathbb{R}^n \). We consider the following linear-quadratic optimal control problem:

\[
\begin{align*}
\text{Maximize } & J(u) = \frac{1}{2}x(T_f)^T Q x(T_f) + w^T x(T_f), \\
\dot{x}(t) & = Ax(t) + Bu(t), \ x(0) = x_0, \\
Hx(T_f) & = g, \\
\alpha \leq u(t) \leq \beta, \ t \in [0, T_f], \ T_f \text{ fix},
\end{align*}
\]

where \( x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T \) is the vector of states at the instant \( t \), \( u(t) = (u_1(t), u_2(t), \ldots, u_m(t))^T \) is the vector of controls at the instant \( t \), which are measurable and locally bounded applications on \( [0, T_f] \), with values in the subset \( \Omega = [\alpha, \beta] \subset \mathbb{R}^m \), \( \alpha, \beta \in \mathbb{R}^m \), \( Q \in \mathcal{M}_n(\mathbb{R}) \) is a symmetric real matrix of order \( n \), \( w \in \mathbb{R}^n \), \( H \in \mathcal{M}_{p,n}(\mathbb{R}) \), \( g \in \mathbb{R}^p \). The symbol \((^T)\) is the transposition operation.

3. Transformation of the LQOC problem into a quadratic optimization problem. By using the Cauchy formula, the solution of the dynamical system (1.2) is given by:

\[
x(t) = F(t)x_0 + \int_0^t F(t)F(s)^{-1}Bu(s)ds, \ t \in [0, T_f],
\]

where \( F(t) \in \mathcal{M}_n(\mathbb{R}) \) is the resolvent, solution of the system:

\[
\dot{F}(t) = AF(t), F(0) = I_n, \ t \in [0, T_f],
\]
where $I_n$ represents the identity matrix of order $n$.

The solution of the system (3) is given by:

$$F(t) = e^{tA}, \ t \in [0, T_f].$$

We define the matrix $\phi(t) \in \mathcal{M}_{n,m} (\mathbb{R})$, the vectors $y, v \in \mathbb{R}^n$ and $\bar{g} \in \mathbb{R}^p$ as follows:

$$\phi(t) = F(T_f)F(t)^{-1}B, \ y = \int_0^{T_f} \phi(t)u(t)dt, \ v = F(T_f)x_0 \text{ and } \bar{g} = g - Hv. \quad (4)$$

Hence, we get

$$x(T_f) = F(T_f)x_0 + \int_0^{T_f} \phi(t)u(t)dt = v + y. \quad (5)$$

By substituting the expression of the solution $x(T_f)$ in the quality criterion (1.1) and the terminal contraints (1.3), we get:

$$J(u(t)) = \frac{1}{2}(v + y)^TQ(v + y) + w^Tv + w^Ty = \frac{1}{2}v^TQv + v^TQy + \frac{1}{2}y^TQy + w^Tv + w^Ty, \quad (6)$$

and

$$Hy = g - Hv = \bar{g}. \quad (7)$$

Therefore, the problem (1) takes the following form:

$$\begin{cases} 
\max_u J(u) = \frac{1}{2}v^TQv + v^TQy + \frac{1}{2}y^TQy + w^Tv + w^Ty, \\
Hy = \bar{g}, \\
\alpha \leq u(t) \leq \beta, \ t \in [0, T_f].
\end{cases} \quad (8)$$

Let us put

$$J_0 = \frac{1}{2}v^TQv + w^Tv, \ d^T = v^TQ + w^T. \quad (9)$$

Then the problem (8) becomes

$$\begin{cases} 
\max_u J(u) = J_0 + d^Ty + \frac{1}{2}y^TQy, \\
Hy = \bar{g}, \\
\alpha \leq u(t) \leq \beta, \ t \in [0, T_f].
\end{cases} \quad (10)$$

3.1. **Discretization.** For a number of discretization sub-intervals $N \in \mathbb{N}^*$ chosen in advance, the step of discretization is calculated $h = \frac{T_f}{N}$. Let $\tau_{j+1} = \tau_j + h$, so we have $[0, T_f] = \bigcup_{j=1}^N [\tau_j, \tau_{j+1}].$

We calculate the matrices $R_j \in \mathcal{M}_{n,m} (\mathbb{R}), \ j = 1, \ldots, N$ as follows:

$$R_j = \int_{\tau_j}^{\tau_{j+1}} \phi(t)dt, \ j = 1, \ldots, N. \quad (11)$$

We put

$$u^j(t) = u(\tau_j) = u^j \in \mathbb{R}^m, \ \forall t \in [0, T_f], \ j = 1, \ldots, N.$$  

Then we have

$$y = \int_0^{T_f} \phi(t)u(t)dt = \sum_{j=1}^N \left( \int_{\tau_j}^{\tau_{j+1}} \phi(t)dt \right) u^j = \sum_{j=1}^N R_j u^j = Ru.$$
and
\[ H y = H R u = \bar{H} u, \]
where
\[ R = (R_1, R_2, \ldots, R_N) \in \mathcal{M}_{n,mN}(\mathbb{R}), \quad \bar{H} = H R \in \mathcal{M}_{p,mN}(\mathbb{R}) \] (12)
and
\[ u = \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^N \end{pmatrix} \in \mathbb{R}^{mN}. \]

By substituting the expression of \( y \) in problem (10), we get the following quadratic programming problem:
\[
\begin{cases}
\max_u J(u) = J_0 + c^T u + \frac{1}{2} u^T D u, \\
\bar{H} u = \bar{g}, \\
a \leq u \leq b,
\end{cases}
\] (13)
where
\[ D = R^T Q R \in \mathcal{M}_{mN}(\mathbb{R}), \quad c = R^T d \in \mathbb{R}^{mN}, \] (14)
and
\[ a = \begin{pmatrix} \alpha \\ \alpha \\ \vdots \\ \alpha \end{pmatrix} \in \mathbb{R}^{mN}, \quad b = \begin{pmatrix} \beta \\ \beta \\ \vdots \\ \beta \end{pmatrix} \in \mathbb{R}^{mN}. \] (15)

3.2. Cauchy Discretization Algorithm. In this section, we summarize the different steps of the Cauchy Discretization Algorithm.

Algorithm (Cauchy Discretization Algorithm)

Input: \( Q, w, A, B, x_0, H, g, \alpha, \beta, T_f \);

Output: The maximum \( J^* \) and the optimal control \( u^* \);

Step 1. Calculate the vectors \( v \) and \( \bar{g} \) with relationship (4);

Step 2. Calculate the value \( J_0 \) and the vector \( d^T \) with relationship (9);

Step 3. Calculate the \((n \times m)\)-matrices \( R_j, j = 1, \ldots, N \) with (11);

Step 4. Calculate the matrices \( R \) and \( \bar{H} \) with (12);

Step 5. Calculate the matrix \( D \), the vectors \( c, a \) and \( b \) with (14)-(15);

Step 6. Solve the quadratic optimization problem (13).

We consider two versions of CDA:
• **CDA1**: we calculate the integral \( I(t) = \int_0^t \phi(t) dt \) symbolically, then we calculate the matrices

\[
R_j = \int_{\tau_j}^{\tau_{j+1}} \phi(t) dt = I(\tau_{j+1}) - I(\tau_j).
\]

• **CDA2**: we calculate the definite integrals \( R_j \) using the following Simpson’s rule:

\[
R_j = \int_{\tau_j}^{\tau_{j+1}} \phi(t) dt \simeq \frac{h}{6} \left( \phi(\tau_j) + 4\phi(\frac{\tau_j + \tau_{j+1}}{2}) + \phi(\tau_{j+1}) \right). \tag{16}
\]

4. **Numerical examples.**

4.1. **Example 1.** Consider the problem of the vehicle moving on a straight line, modelled by the control system

\[
\frac{d^2x}{dt^2}(t) = u(t), \quad x(0) = \frac{dx}{dt}(0) = 0, \forall t \in [0,T_f], \tag{17}
\]

subject to the constraints:

\[
x_1(T_f) - 2x_2(T_f) = \frac{1}{2} \text{ and } | u(t) | \leq 1, \tag{18}
\]

with \( x_1 = x \) and \( x_2 = \dot{x} \).

The equations (17) and (18) are equivalent to

\[
\begin{aligned}
\dot{x}_1(t) &= x_2(t), \\
\dot{x}_2(t) &= u(t), \\
x_1(T_f) - 2x_2(T_f) &= \frac{1}{2}, \\
x_1(0) &= x_2(0) = 0, \quad -1 \leq u(t) \leq 1, \quad t \in [0,T_f].
\end{aligned} \tag{19}
\]

We wish for a fixed final time \( T_f = 2 \) s, maximize the following quality criterion:

\[
J(u(t)) = x_1(T_f)^2 + x_2(T_f)^2.
\]

The problem is formulated as follows:

\[
\begin{aligned}
\text{max } J(u(t)) &= x_1(T_f)^2 + x_2(T_f)^2, \\
\dot{x}_1(t) &= x_2(t), \\
\dot{x}_2(t) &= u(t), \\
x_1(T_f) - 2x_2(T_f) &= \frac{1}{2}, \\
x_1(0) &= x_2(0) = 0, \quad -1 \leq u(t) \leq 1, \quad t \in [0,T_f].
\end{aligned} \tag{20}
\]
4.1.1. **Resolution by the Analytical Method.** In order to solve the problem (20), we first apply the Pontryagin’s maximum principle. The Hamiltonian of the problem (20) is given by:

\[
H(x(t), p(t), u(t)) = p_1(t) \dot{x}_1(t) + p_2(t) \dot{x}_2(t) = p_1(t)x_2(t) + p_2(t)u(t).
\]

The adjoint vector \( p(t) = (p_1(t), p_2(t)) \) is a solution of the following system:

\[
\begin{cases}
\dot{p}_1(t) = 0, \\
\dot{p}_2(t) = -p_1(t), \quad t \in [0, T_f].
\end{cases}
\]

From the system (21), we deduce that

\[
\begin{cases}
p_1(t) = \lambda_1, \\
p_2(t) = -\lambda_1 t + \lambda_2, \quad \lambda_1, \lambda_2 \in \mathbb{R}.
\end{cases}
\]

The maximum of the Hamiltonian is given by:

\[
H(x(t), p(t), u^*(t)) = \max_{-1 \leq u(t) \leq 1} H(x(t), p(t), u(t)) = p_1(t)x_2(t) + \max_{-1 \leq u(t) \leq 1} [p_2(t)u(t)].
\]

The control maximizing the Hamiltonian is:

\[
u^*(t) = \text{sign}(p_2(t)).
\]

Since the optimal control is equal to the sign of \( p_2(t) \), the switching time \( t_c \) is given by the root of the equation \( p_2(t) = 0 \). In order to choose the optimal strategy, consider the following possible strategies:

- **Strategy 1:** \( u(t) = 1, \quad \forall t \in [0, T_f] \);
- **Strategy 2:** \( u(t) = -1, \quad \forall t \in [0, T_f] \);
- **Strategy 3:** \( u(t) = 1 \) for \( t \in [0, t_c] \) then \( u(t) = -1 \), for \( t \in [t_c, T_f] \);
- **Strategy 4:** \( u(t) = -1 \) for \( t \in [0, t_c] \) then \( u(t) = 1 \), for \( t \in [t_c, T_f] \).

**Strategy 1:**

From the dynamical system of problem (20), we have \( \dot{x}_2(t) = 1 \), which implies

\[
\begin{cases}
x_2(t) = t + \alpha_1, \\
x_1(t) = \frac{t^2}{2} + \alpha_1 t + \alpha_2, \quad \alpha_1, \alpha_2 \in \mathbb{R}.
\end{cases}
\]

Using the initial conditions, we obtain

\[
\begin{cases}
x_2(t) = t, \\
x_1(t) = \frac{t^2}{2}.
\end{cases}
\]

For \( t = 2 \), we will have \( x_1(2) - 2x_2(2) = -2 \neq \frac{1}{2} \), therefore strategy 1 is not feasible.

**Strategy 2:**

We have \( \dot{x}_2(t) = -1 \), which implies

\[
\begin{cases}
x_2(t) = -t + \beta_1, \\
x_1(t) = -\frac{t^2}{2} + \beta_1 t + \beta_2, \quad \beta_1, \beta_2 \in \mathbb{R}.
\end{cases}
\]
Using the initial conditions, we obtain:

\[
\begin{align*}
    x_2(t) &= -t, \\
    x_1(t) &= -\frac{t^2}{2},
\end{align*}
\]  

(27)

For \( t = 2 \), we will have \( x_1(2) - 2x_2(2) = 2 \neq \frac{1}{2} \), therefore strategy 2 is not feasible.

**Strategy 3:**

The equations of the first piece of trajectories, when \( u(t) = 1 \), \( t \in [0, t_c[ \), are:

\[
\begin{align*}
    x_2(t) &= t, \\
    x_1(t) &= \frac{t^2}{2}.
\end{align*}
\]  

(28)

The equations of the second piece of trajectories, when \( u(t) = -1 \), \( t \in [t_c, T_f] \), are:

\[
\begin{align*}
    x_2(t) &= -t + c_1, \\
    x_1(t) &= -\frac{t^2}{2} + c_1 t + c_2, \quad c_1, c_2 \in \mathbb{R}.
\end{align*}
\]  

(29)

At the point of intersection of the two pieces, we obtain:

\[ c_1 = 2t_c, \quad c_2 = -t_c^2. \]

Let us compute \( t_c \) such that \( x_1(2) - 2x_2(2) = \frac{1}{2} \).

This last condition leads to the following equation:

\[ t_c^2 - \frac{3}{2} = 0, \]  

(30)

which has the solution: \( t_c = \sqrt{\frac{3}{2}} \).

By replacing the values of \( c_1 \) and \( c_2 \) in the system (29), we obtain the following optimal value of the quality criterion:

\[ J(u) = x_1(T_f)^2 + x_2(T_f)^2 \approx 2.159185. \]

**Strategy 4:**

The equations of the first piece of the trajectories, when \( u(t) = -1, t \in [0, t_c[ \) are:

\[
\begin{align*}
    x_2(t) &= -t, \\
    x_1(t) &= -\frac{t^2}{2}.
\end{align*}
\]  

(31)

The equations of the second piece of trajectories, when \( u(t) = 1 \), \( t \in [t_c, T_f] \), are:

\[
\begin{align*}
    x_2(t) &= t + d_1, \\
    x_1(t) &= \frac{t^2}{2} + d_1 t + d_2, \quad d_1, d_2 \in \mathbb{R}.
\end{align*}
\]  

(32)

At the point of intersection of the two pieces, we obtain:

\[ d_1 = -2t_c, \quad d_2 = t_c^2. \]

Let us compute \( t_c \) such that \( x_1(2) - 2x_2(2) = \frac{1}{2} \).

This last condition gives:

\[ t_c^2 - \frac{5}{2} = 0, \]  

(33)
which has the solution: \( t_c = \sqrt{\frac{5}{2}} \).

By substituting the values of \( d_1 \) and \( d_2 \) in the system (32), we obtain:
\[
J(u) = x_1(T_f)^2 + x_2(T_f)^2 \simeq 4.679891.
\]

Note that the optimal value of the quality criterion given by strategy 4 is greater than the one found by strategy 3, which is of the bang-bang type. We have therefore determined the optimal trajectory that maximizes \( J(u) \) and satisfies the terminal constraint \( x_1(2) - 2x_2(2) = \frac{1}{2} \).

\[
u^*(t) = \begin{cases} 
-1, & t \in [0, \sqrt{\frac{5}{2}}]; \\
1, & t \in [\sqrt{\frac{5}{2}}, 2];
\end{cases}
\quad \text{and} \quad J(u^*) \simeq 4.679891. \tag{34}
\]

4.1.2. Numerical resolution by the Euler Discretization Method. For a number of sub-intervals \( N \) chosen in advance, we will have the step of discretization \( h = \frac{T_f}{N} \) and the following moments:
\[
0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T_f.
\]

The application of the Euler scheme for solving boundary value problems gives us the following concave quadratic programming programming problem:
\[
\begin{cases}
\text{Maximize } J(u) = x_1(T_f)^2 + x_2(T_f)^2, \\
x_1(t_{i+1}) = x_1(t_i) + hx_2(t_i), \quad x_1(0) = 0, \quad i = 0, 1, \ldots, N-1, \\
x_2(t_{i+1}) = x_2(t_i) + hu(t_i), \quad x_2(0) = 0, \quad i = 0, 1, \ldots, N-1, \\
x_1(t_N) - 2x_2(t_N) = \frac{1}{2}, \\
-1 \leq u(t_i) \leq 1, \quad i = 0, 1, \ldots, N-1.
\end{cases}
\tag{35}
\]

We developed an implementation with Matlab2012b that solves the problem (35) with the function “fmincon”. This function finds the minimum of non-linear programming problems with linear and non-linear constraints. We have used two algorithms for solving the optimization problem: Sequential Quadratic Programming (SQP) [15] and the Active-Set Method (ASM). The matlab code is executed on a Dell PC with a Core i5 M560 microprocessor, 2.67Ghz, RAM 4 GB working under Windows 7. The numerical results for different values of \( N \) are presented in Table 1, where \( J(u), \text{CPU} \) and \( \text{Error} \) represent respectively the approximate optimal value of the quality criterion, the execution time and the absolute value of the difference between the approximate maximum and the analytical maximum \( (\text{Error} = |J(u) - J(u^*)|) \).

4.1.3. Numerical resolution by the Cauchy discretization method. Let us write the linear-quadratic optimal control problem (20) in matrix form:
\[
\begin{cases}
\text{max } J(u(t)) = \frac{1}{2}x(T_f)^TQx(T_f), \\
\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = 0, \\
Hx(T_f) = \frac{1}{2}, \quad -1 \leq u(t) \leq 1, \quad t \in [0, T_f], \quad T_f = 2,
\end{cases}
\tag{36}
\]

where
\[
x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, \quad x(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},
\]

\[
\text{Error} = |J(u) - J(u^*)|.
\]
The solution of the dynamical system of problem (36) at the final time $T_f$ is given by:

$$x(2) = F(2)x_0 + \int_0^2 F(2)(F(t))^{-1}Bu(t)dt,$$

where $F(t)$ is the solution of the system:

$$\dot{F}(t) = AF(t), \quad F(0) = I_2, \quad t \in [0, 2].$$

We have

$$\phi(t) = F(2)(F(t))^{-1}B = \begin{pmatrix} 2 - t \\ 1 \end{pmatrix}$$

and

$$y = \int_0^2 \phi(t)u(t)dt.$$

Hence

$$x(2) = \int_0^2 \phi(t)u(t)dt = y.$$

By replacing the solution (38) in the problem (36), we get the following problem:

$$\begin{align*}
\max J(u) &= \frac{1}{2}y^TQy, \\
Hy &= \frac{1}{2}, \\
-1 &\leq u(t) \leq 1, \quad t \in [0, 2].
\end{align*}$$

**Discretization**

Let $\tau_{j+1} = \tau_j + h$, so we have $[0, T_f] = \bigcup_{j=1}^N [\tau_j, \tau_{j+1}]$.

We calculate the vectors $R_j, \ j = 1, \ldots, N$ as follows:

$$R_j = \int_{\tau_j}^{\tau_{j+1}} \phi(t)dt = \int_{\tau_j}^{\tau_{j+1}} \begin{pmatrix} 2 - t \\ 1 \end{pmatrix} dt = \begin{pmatrix} -\frac{h}{2}(\tau_j + \tau_{j+1}) + 2 \\ 1 \end{pmatrix}, \quad j = 1, \ldots, N.$$

---

Table 1. Numerical simulation results of EDM for Example 1

| N    | J(u) (EDM(SQP)) | CPU (EDM(SQP)) | Error (EDM(SQP)) | J(u) (EDM(ASM)) | CPU (EDM(ASM)) | Error (EDM(ASM)) |
|------|----------------|---------------|-----------------|----------------|---------------|-----------------|
| 4    | 0.953125       | 0.45          | 3.73E+00        | 0.953125       | 1.03          | 3.73E+00        |
| 10   | 2.588281       | 0.09          | 2.09E+00        | 2.588281       | 0.12          | 2.09E+00        |
| 30   | 3.865982       | 0.15          | 8.14E-01        | 3.865982       | 0.25          | 8.14E-01        |
| 50   | 4.176861       | 0.43          | 5.03E-01        | 4.176861       | 0.54          | 5.03E-01        |
| 70   | 4.316335       | 0.91          | 3.64E-01        | 4.316335       | 1.58          | 3.64E-01        |
| 100  | 4.422900       | 1.99          | 2.57E-01        | 4.422900       | 2.48          | 2.57E-01        |
| 200  | 4.549882       | 10.30         | 1.30E-01        | 4.549882       | 12.45         | 1.30E-01        |
| 400  | 4.614494       | 51.99         | 6.54E-02        | 4.614494       | 46.91         | 6.54E-02        |
| 600  | 4.636198       | 161.93        | 4.37E-02        | 4.636198       | 144.54        | 4.37E-02        |
| 800  | 4.647087       | 375.59        | 3.28E-02        | 4.647087       | 379.60        | 3.28E-02        |
| 1000 | 4.653632       | 696.03        | 2.63E-02        | 4.653632       | 865.84        | 2.63E-02        |

and

$$B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad H = (1, -2).$$
We set $u^j = u(t) = u(\tau_j)$, $t \in [\tau_j, \tau_{j+1}]$, $j = 1, \ldots, N$. Thus
\[
y = \int_0^{T_f} \phi(t)u(t)dt = \sum_{j=1}^{N} \left( \int_{\tau_j}^{\tau_{j+1}} \phi(t)dt \right) u^j = \sum_{j=1}^{N} R_j u^j = Ru,
\]
where $u = (u^1, u^2, \ldots, u^N)^T \in \mathbb{R}^N$ and $R = (R_1, R_2, \ldots, R_N) \in \mathcal{M}_{n,N}$. Therefore, we get the following quadratic programming problem:
\[
\begin{cases}
\max_u J(u) = \frac{1}{2} u^T Du,
\bar{H}u = \frac{1}{2}, \\
-1 \leq u \leq 1,
\end{cases}
\]
where $D = R^T QR \in \mathcal{M}_{n,N}$, $\bar{H} = HR \in \mathbb{R}^N$.

For example, if the number of discretization sub-intervals $N = 2$, then we obtain the following concave quadratic optimization problem:
\[
\begin{cases}
\max_u J(u) = 3.25u_1^2 + 3.5u_1u_2 + 1.25u_2^2, \\
-0.5u_1 - 1.5u_2 = \frac{1}{2}, \\
-1 \leq u_1, u_2 \leq 1.
\end{cases}
\]

We have solved the concave quadratic program (40) for different values of $N$ with the active-set method implemented in MATLAB2012b executed on a Dell PC having a Core i5 M560 microprocessor, 2.670Ghz, RAM 4 GO. We report in Table 2, for the two variants CDA1 and CDA2, the execution time in seconds (CPU), the approximate maximum ($J(u)$) and the approximation error (Error), for different values of $N$.

The optimal control found by the analytical method and the approximate optimal control for $N = 1000$ found by CDA are plotted in Figure 1. The CPU time, and the approximation error for $N \geq 30$ are plotted in Figure 2.

From Tables 1, 2 and Figure 2, we deduce that:

| N  | $J(u)$  | CPU  | Error    | $J(u)$  | CPU  | Error    |
|----|---------|------|----------|---------|------|----------|
| 4  | 4.494898| 0.01 | 1.85E-01 | 4.494898| 0.05 | 1.85E-01 |
| 10 | 4.658000| 0.01 | 2.19E-02 | 4.658000| 0.07 | 2.19E-02 |
| 30 | 4.674355| 0.03 | 5.54E-03 | 4.674355| 0.21 | 5.54E-03 |
| 50 | 4.677464| 0.09 | 2.43E-03 | 4.677464| 0.45 | 2.43E-03 |
| 70 | 4.678780| 0.11 | 1.11E-03 | 4.678780| 0.54 | 1.11E-03 |
| 100| 4.679761| 0.25 | 1.30E-04 | 4.679761| 0.81 | 1.30E-04 |
| 200| 4.679830| 0.96 | 6.13E-05 | 4.679830| 8.37 | 6.13E-05 |
| 400| 4.679865| 10.73| 2.67E-05 | 4.679865| 10.08| 2.67E-05 |
| 600| 4.679876| 31.06| 1.52E-05 | 4.679876| 29.00| 1.52E-05 |
| 800| 4.679882| 72.20| 9.43E-06 | 4.679882| 80.56| 9.43E-06 |
| 1000|4.679886|160.93|5.97E-06 |4.679886|178.74|5.97E-06 |

- For $N \leq 50$, the solution obtained by EDM is of bang-bang-singular type, then it is not optimal.
For $50 < N \leq 200$, the Euler discretization method is fast. However, it gives an approximate value which is far from the optimal value obtained by the analytical method.

For large values of $N$ ($200 < N \leq 1000$), the Euler discretization method converges slowly to an approximate optimal value which is close to the analytical optimal one.

For $N = 100$, the Cauchy discretization method finds an approximate maximum with an accuracy of 3 exact decimals, an approximation error equal to $1.3 \times 10^{-4}$ and an execution time of $0.25$ s; and for $200 \leq N \leq 1000$ it finds an approximate maximum with an accuracy of four exact decimals, an error varying between $6.13 \times 10^{-5}$ and $5.97 \times 10^{-6}$, and an execution time varying
4.2. Example 2. Consider two material points whose equations of motion are
\[
\begin{align*}
\ddot{y}_1(t) &= u_1(t), \quad y_1(0) = \dot{y}_1(0) = y_1(0) = 0, \\
\ddot{y}_2(t) &= u_2(t), \quad y_2(0) = 5, \quad \dot{y}_2(0) = y_2(0) = 0, \\
\ddot{y}_1(T_f) &= \ddot{y}_2(T_f) = -6, \quad |u_1| \leq 3, \quad |u_2| \leq 3, \quad t \in [0,T_f], \quad T_f = 4.
\end{align*}
\]  
(42)

We set:
\[
x_1 = y_1, \quad x_2 = \dot{y}_1, \quad x_3 = \ddot{y}_1, \quad x_4 = y_2, \quad x_5 = \dot{y}_2, \quad x_6 = \ddot{y}_2.
\]

The accelerations of the two material points at the final time $T_f$ will verify
\[
x_3(T_f) = x_6(T_f) = -6
\]
and the following objective functional will be minimized:
\[
J(u) = (x_1(T_f) - x_4(T_f))^2 + (x_2(T_f) - x_5(T_f))^2 \\
+ 2x_1(T_f) - 324x_2(T_f) + 64x_4(T_f) + 37x_5(T_f).
\]

We obtain the following convex linear-quadratic optimal control problem [11]:
\[
\begin{align*}
\min J(u) &= (x_1(T_f) - x_4(T_f))^2 + (x_2(T_f) - x_5(T_f))^2 \\
&\quad + 2x_1(T_f) - 324x_2(T_f) + 64x_4(T_f) + 37x_5(T_f), \\
\dot{x}_1(t) &= x_2(t), \\
\dot{x}_2(t) &= x_3(t), \\
\dot{x}_3(t) &= u_1(t), \\
\dot{x}_4(t) &= x_5(t), \\
\dot{x}_5(t) &= x_6(t), \\
\dot{x}_6(t) &= u_2(t), \\
x_1(0) &= x_2(0) = x_3(0) = 0, \quad x_4(0) = 5, \quad x_5(0) = x_6(0) = 0, \\
x_3(T_f) &= x_6(T_f) = -6, \\
|u_1| &\leq 3, \quad |u_2| \leq 3, \quad t \in [0,T_f], \quad T_f = 4.
\end{align*}
\]  
(43)

Let us write the problem (43) in matrix form. We get the following problem:
\[
\begin{align*}
\min J(u(t)) &= \frac{1}{2} x(T_f)^T Q x(T_f) + w^T x(T_f), \\
\dot{x}(t) &= Ax(t) + Bu(t), \\
H x(T_f) &= g, \quad |u_1| \leq 3, \quad |u_2| \leq 3, \quad t \in [0,T_f], \quad T_f = 4.
\end{align*}
\]  
(44)

between 0.96 s and 160.93 s, while the Euler discretization method finds for $N = 1000$ an approximate optimal value with an accuracy of one exact decimal, an error equal to $2.6 \times 10^{-2}$, and an execution time of 696.03 s with SQP and 865.84 s with ASM. Therefore, the Cauchy discretization method is more accurate and faster than the classical Euler discretization method for this example.
where
\[
x(t) = \begin{pmatrix}
x_1(t) \\
x_2(t) \\
x_3(t) \\
x_4(t) \\
x_5(t) \\
x_6(t)
\end{pmatrix}, \quad x(0) = \begin{pmatrix}
0 \\
0 \\
0 \\
5 \\
0 \\
0
\end{pmatrix}, \quad Q = \begin{pmatrix}
2 & 0 & 0 & -2 & 0 & 0 \\
0 & 2 & 0 & 0 & -2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-2 & 0 & 0 & 2 & 0 & 0 \\
0 & -2 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}, \quad A = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}, \quad B = \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix},
\]
and
\[
H = \begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}, \quad g = \begin{pmatrix}
-6 \\
-6
\end{pmatrix}, \quad w = (2 \quad -324 \quad 0 \quad 64 \quad 37 \quad 0)^T.
\]

The optimal cost found in [11] is \( J(u^*) = -173.999843 \).

The numerical results for this example are shown in Tables 3 and 4. The CPU time, and the approximation error for \( N \geq 30 \) are plotted in Figure 3.

| \( N \) | EDM(SQP) | EDM(ASM) |
|---|---|---|
| \( J(u) \) | CPU | Error | \( J(u) \) | CPU | Error |
| 4 | -807.000006 | 42 Ms | 6.33E+02 | -807.000000 | 78 Ms | 6.33E+02 |
| 10 | -6.020544 | 4 Ms | 1.68E+02 | -6.021052 | 78 Ms | 6.33E+02 |
| 30 | -180.145998 | 17 Ms | 6.15E+00 | -180.145998 | 24 Ms | 6.15E+00 |
| 50 | -187.685200 | 74 Ms | 1.37E+01 | -187.685200 | 66 Ms | 1.37E+01 |
| 70 | -187.041677 | 80 Ms | 1.30E+01 | -187.041677 | 85 Ms | 1.30E+01 |
| 100 | -184.758275 | 504 s | 1.08E+01 | -184.758275 | 924 s | 1.08E+01 |
| 200 | -180.485036 | 37.17 s | 6.49E+00 | -180.485036 | 36.75 s | 6.49E+00 |
| 400 | -177.524433 | 293.36 s | 3.52E+00 | -177.523813 | 324.39 s | 3.52E+00 |
| 600 | -176.412810 | 1075.55 s | 2.41E+00 | -176.411448 | 1284.80 s | 2.41E+00 |
| 800 | -175.833379 | 2692.90 s | 1.83E+00 | -175.831972 | 3712.96 s | 1.83E+00 |
| 1000 | -175.478114 | 5611.01 s | 1.48E+00 | -175.477201 | 8598.05 s | 1.48E+00 |

From Tables 3, 4 and Figure 3, we remark that the Cauchy discretization method found the maximum of the convex LQOC problem with a great accuracy for \( N = 4 \) and \( N \geq 100 \). For \( N = 4 \), the execution time of CDA is equal to 0.29 s and for \( 100 \leq N \leq 1000 \), the CPU time varies between 0.98 s and 2354.52 s. The Euler discretization method found for \( N = 1000 \) an approximate maximum with an approximation error equal to 1.48 and an execution time of 5611.01 s with SQP and 8598.05 s with ASM. Therefore, CDA is more accurate and faster than EDM for this example. Moreover, we remark that the sequential quadratic programming algorithm is more efficient than the active-set method for solving the optimization problem obtained with the Euler discretization method, which has 8000 variables and 6002 constraints.
Table 4. Numerical simulation results of CDA for Example 2

| N   | J(u)  | CPU  | Error   | J(u)  | CPU  | Error   |
|-----|-------|------|---------|-------|------|---------|
| 4   | -174.000000 | 0.29 | 9.95E-13 | -174.000000 | 0.56 | 0.00E+00 |
| 10  | -161.832000  | 0.02 | 1.22E+01 | -161.832000  | 0.11 | 1.22E+01 |
| 30  | -172.676444  | 0.07 | 1.32E+00 | -172.676444  | 0.33 | 1.32E+00 |
| 50  | -173.524339  | 0.19 | 4.76E-01 | -173.524339  | 0.61 | 4.76E-01 |
| 70  | -173.757431  | 0.44 | 2.43E-01 | -173.757431  | 0.98 | 2.43E-01 |
| 100 | -174.000000  | 0.98 | 3.01E-12 | -174.000000  | 1.81 | 5.00E-12 |
| 200 | -174.000000  | 8.02 | 4.01E-12 | -174.000000  | 10.29 | 6.00E-12 |
| 400 | -174.000000  | 83.60 | 3.00E-11 | -174.000000  | 90.35 | 9.00E-11 |
| 600 | -174.000000  | 354.27 | 2.10E-11 | -174.000000  | 359.11 | 6.40E-11 |
| 800 | -174.000000  | 996.67 | 6.20E-11 | -174.000000  | 1045.79 | 1.56E-10 |
| 1000| -174.000000 | 2354.52 | 3.01E-12 | -174.000000 | 2320.88 | 4.01E-12 |

Figure 3. CPU time and Error in terms of N for Example 2

4.3. Example 3. Let us consider the following concave LQOC problem [7]:

\[
\begin{align*}
\max J(u(t)) &= x_1(2)^2 + x_2(2)^2, \\
\dot{x}_1(t) &= x_2(t), \\
\dot{x}_2(t) &= u(t), \\
x_1(0) &= x_2(0) = 1, \\
-2 &\leq u(t) \leq 2, \ t \in [0, 2].
\end{align*}
\]

The optimal value of the quality criterion [7]: \( J(u^*) = 74 \).

The numerical results of this example are shown in Tables 5 and 6. The CPU time, and the approximation error for \( N \geq 30 \) are plotted in Figure 4.
Table 5. Numerical simulation results of EDM for Example 3

| N  | EDM(SQP) | EDM(ASM) |
|----|----------|----------|
|    | J(u)     | CPU      | Error   | J(u)     | CPU      | Error   |
| 4  | 32.000000 | 0.45     | 4.20E+01| 32.000000| 0.85     | 4.20E+01|
| 10 | 53.422400 | 0.49     | 2.06E+01| 53.422400| 0.93     | 2.06E+01|
| 30 | 66.485116 | 0.54     | 7.51E+00| 66.485116| 1.02     | 7.51E+00|
| 50 | 69.407818 | 0.66     | 4.59E+00| 69.407818| 1.22     | 4.59E+00|
| 70 | 70.694043 | 0.88     | 3.31E+00| 70.694043| 1.51     | 3.31E+00|
| 100| 71.672177 | 1.26     | 2.33E+00| 71.672177| 2.01     | 2.33E+00|
| 200| 72.828072 | 2.84     | 1.17E+00| 72.828072| 4.14     | 1.17E+00|
| 400| 73.412022 | 12.25    | 5.88E-01| 73.412022| 16.85    | 5.88E-01|
| 600| 73.607566 | 42.80    | 3.92E-01| 73.607566| 56.17    | 3.92E-01|
| 800| 73.705506 | 120.33   | 2.94E-01| 73.705506| 154.07   | 2.94E-01|
| 1000|73.764324 | 284.73   | 2.36E-01| 73.764324| 360.15   | 2.36E-01|

From Tables 5, 6 and Figure 4, we remark that the Cauchy discretization method successfully found the analytical maximum of the concave LQOC problem without terminal constraints (45) for all the values of N, with a CPU time varying between 0.01 s and 144.16 s, while the Euler discretization method found for N = 1000 an approximate optimal value with an approximation error equal to $2.36 \times 10^{-1}$ and an execution time of 284.73 s with SQP and 360.15 s with ASM. Therefore, CDA is more accurate and faster than EDM for this example.
4.4. Example 4. Let us consider the following indefinite LQOC problem:

\[
\begin{aligned}
\max J(u(t)) &= x_1(T)^2 - x_2(T)^2, \\
\dot{x}_1(t) &= x_2(t), \\
\dot{x}_2(t) &= u(t), \\
x_1(T) - 2x_2(T) &= \frac{1}{2}, \\
x_1(0) = x_2(0) &= 0, -1 \leq u(t) \leq 1, \quad t \in [0, T].
\end{aligned}
\] (46)

The optimal value of the quality criterion found by the analytical method is: 
\[ J(u^*) \simeq 1.978113. \]

The numerical results of this example are shown in Tables 7 and 8. The CPU time, and the approximation error for \( N \geq 30 \) are plotted in Figure 5.

From Tables 7, 8 and Figure 5, we remark that CDA found for \( N = 1000 \) the maximum of the indefinite linear-quadratic optimal control problem with an accuracy of five exact decimals, an error equal to \( 3.09 \times 10^{-6} \), and an execution time of 157.80 s, while EDM found for \( N = 1000 \) an approximate maximum with an error equal to \( 1.3 \times 10^{-2} \) and an execution time of 820.38 s with SQP and 1180.43 s with ASM. Therefore, CDA is more accurate and faster than EDM for this example. Moreover, EDM(SQP) is faster than EDM(ASM) and CDA using the Simpson’s rule is faster than CDA using the symbolic technique for computing definite integrals.

Remark 1. Note that the quadratic program obtained after the discretization phase of CDA has \( mN \) variables and \( p \) constraints, while the quadratic program obtained with EDM has \( (m + n)N \) variables and \( nN + p \) constraints. For example, the quadratic program of exemple 2 obtained with CDA has 2000 variables and 2 constraints and the quadratic programming problem obtained with EDM has 8000 variables and 6002 constraints. Therefore, when the number of state variables \( n \),
Table 7. Numerical simulation results of EDM for Example 4

| N   | EDM(SQP) 𝐽(𝐮) | EDM(SQP) CPU error | EDM(ASM) 𝐽(𝐮) | EDM(ASM) CPU error |
|-----|----------------|-------------------|----------------|-------------------|
| 4   | 0.171875       | 0.46              | 0.171875       | 1.03              |
| 10  | 0.922969       | 0.07              | 0.922969       | 0.08              |
| 30  | 1.560700       | 0.17              | 1.560700       | 0.16              |
| 50  | 1.719317       | 0.60              | 1.719317       | 0.53              |
| 70  | 1.790821       | 1.12              | 1.790821       | 1.00              |
| 100 | 1.845588       | 2.58              | 1.845588       | 2.06              |
| 200 | 1.910993       | 12.49             | 1.910993       | 10.75             |
| 300 | 1.944331       | 60.91             | 1.944331       | 61.13             |
| 500 | 1.955538       | 187.13            | 1.955538       | 176.01            |
| 700 | 1.975251       | 738.55            | 1.975251       | 79.62             |
| 1000| 1.978046       | 157.80            | 1.978046       | 177.02            |

Table 8. Numerical simulation results of CDA for Example 4

| N   | CDA1 𝐽(𝐮) | CDA1 CPU error | CDA2 𝐽(𝐮) | CDA2 CPU error |
|-----|-----------|----------------|-----------|----------------|
| 4   | 1.882653  | 0.01           | 9.55E-02  | 0.06           |
| 10  | 1.966800  | 0.01           | 1.13E-02  | 0.07           |
| 30  | 1.975251  | 0.03           | 2.86E-03  | 0.21           |
| 50  | 1.976858  | 0.08           | 1.25E-03  | 0.36           |
| 70  | 1.977538  | 0.11           | 5.74E-04  | 0.58           |
| 100 | 1.978046  | 0.23           | 6.72E-05  | 0.80           |
| 200 | 1.978081  | 1.05           | 3.17E-05  | 2.92           |
| 400 | 1.978099  | 10.23          | 1.38E-05  | 10.02          |
| 600 | 1.978105  | 30.87          | 7.86E-06  | 28.47          |
| 800 | 1.978108  | 73.76          | 4.88E-06  | 79.62          |
| 1000| 1.978110  | 157.80         | 3.09E-06  | 177.02         |

the number of control variables $m$ and the number of terminal constraints $p$ are large, the continuous problem cannot be solved analytically and the superiority of the Cauchy discretization method over the Euler discretization one in terms of CPU time and accuracy will increase.

**Remark 2.** Although the active-set method is an algorithm which finds local solutions of non-convex quadratic programming problems, it has successfully found the global optimum of the non-convex quadratic programs of examples 1, 3 and 4 obtained with the Cauchy discretization algorithm. However, ASM can give local solutions which are not global for other non-convex test-problems not considered in this work. So in order to find a good approximate global optimal solution for the non-convex quadratic programming problems of the form (13), we can use the sequential linear programming algorithm developed in [4] for solving non-convex quadratic programs.

5. **Conclusion.** In this work, we have proposed a new method for solving linear-quadratic optimal control problems. First a change of variable is made using the
Cauchy formula, after that a discretization phase is performed and a quadratic programming problem is obtained. Finally, this problem is solved using the active-set method. The proposed method can find approximate solutions for convex as well as non-convex problems. In order to compare our approach with the Euler discretization method, we have developed an implementation with MATLAB. The obtained numerical simulation results on four convex and non-convex linear-quadratic optimal control problems arising in mechanics show that the new direct method converges fastly to the optimal value of the quality criterion found by the analytical method. In a future work, we will apply the algorithm developed in [4] to efficiently solve the non-convex quadratic programming problems obtained by the discretization methods presented in this work. Furthermore, we will adapt the proposed algorithm for solving linear-quadratic optimal control problems, where the matrices of the linear dynamical system depend on the time.

REFERENCES

[1] J. Awerjewicz, *Modeling, Simulation and Control of Non-Linear Engineering Dynamical Systems, State-of-the-Art, Perspectives and Applications*, Springer, Heidelberg, Germany, 2008.

[2] M. Aliane, N. Moussouni and M. Bentobache, Optimal control of a rectilinear motion of a rocket, *Statistics, Optimization & Information Computing*, 8 (2020), 281–295.

[3] M. Aliane, N. Moussouni and M. Bentobache, Non-linear optimal control of the heel angle of a rocket, *The 6th International Conference on Control, Decision and Information Technologies CODIT’19, Paris, France, (2019), 756–760.

[4] M. Bentobache, M. Telli and A. Mokhtari, A sequential linear programming algorithm for continuous and mixed-integer non-convex quadratic programming, *Optimization of Complex Systems: Theory, Models, Algorithms and Applications, WCGO 2019, Advances in Intelligent Systems and Computing*, Springer, Cham, 991 (2020), 26–36.

[5] M. O. Bibi and M. Bentobache, A hybrid direction algorithm for solving linear programs, *International Journal of Computer Mathematics*, 92 (2015), 201–216.

[6] L. D. Duncan, *Basic Considerations in the Development of an Unguided Rocket Trajectory Simulation Model*, Technical report No 5076, Atmospheric Sciences Laboratory, United States Army Electronics Command, 1966.
[7] A. Chinchuluun, R. Enkhbat, P. M. Pardalos, A novel approach for non-convex optimal control problems, *Optimization*, 58 (2009), 781–789.

[8] R. Gabasov, F. M. Kirillova and S. V. Prischepova, *Optimal Feedback Control*, Springer-Verlag, London, 1995.

[9] G. R. Rose, *Numerical Methods for Solving Optimal Control Problems*, Master’s Thesis, University of Tennessee, Knoxville, 2015.

[10] P. Howlett, The optimal control of a train, *Annals of Operations Research*, 98 (2000), 65–87.

[11] N. Khimoum and M. O. Bibi, Primal-dual method for solving a linear-quadratic multi-input optimal control problem, *Optimization Letters*, 14 (2019), 653–669.

[12] K. Louadj, P. Spiteri, F. Demim, M. Aidene, A. Nemra and F. Messine, Application optimal control for a problem aircraft flight, *Engineering Science and Technology Review*, 11 (2018), 156–164.

[13] N. Moussouni and M. Aidene, An algorithm for optimization of cereal output, *Acta Applicandae Mathematicae*, 11 (2011), 113–127.

[14] N. Moussouni and M. Aidene, Optimization of cereal output in presence of locusts, *An International Journal of Optimization and Control: Theories & Applications*, 6 (2016), 1–10.

[15] O. Oukacha, *Direct Method for the Optimization of Control Problems*, PhD Dissertation, University of Tizi-Ouzou, Algeria, 2016 (in French).

[16] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience Publisher, New York, 1962.

[17] E. Trélat, *Optimal Control: Theory and Applications*, Vuibert, Paris, France, 2005 (in French).

[18] J. Zhu and R. Zeng, A mathematical formulation for optimal control of air pollution, *Science in China*, 46 (2003), 994–1002.

[19] M. A. Zaitri, M. O. Bibi and M. Bentobache, A hybrid direction algorithm for solving optimal control problems, *Cogent Mathematics & Statistics*, 6 (2019), 1–12.

Received April 2020; 1st revision August 2020; Final revision December 2020.

E-mail address: m.aliane@lagh-univ.dz
E-mail address: m.bentobache@lagh-univ.dz
E-mail address: nmoussouni@yahoo.fr
E-mail address: philippe.marthon@enseeiht.fr