An Augmented Lagrangian Method for Optimal Control of Continuous Time DAE Systems

Marco Aurelio S. de Aguiar
Dept. of Automation and Systems
UFSC – Brazil
Email: m.aguiar@posgrad.ufsc.br

Eduardo Camponogara
Dept. of Automation and Systems
UFSC – Brazil
Email: eduardo.camponogara@ufsc.br

Bjarne Foss
Dept. of Engineering Cybernetics
NTNU – Norway
email: bjarne.foss@ntnu.no

***Abstract*—The abstract goes here.**

I. Introduction

In the optimization field, the Augmented Lagrangian method obtains a solution to a Constrained Optimization Problem (COP) by solving a sequence of Unconstrained Optimization Problems (UOPs) that, in general, are more easily solved. Depending on the problem structure, each UOP can be divided into subproblems that can be solved in a distributed fashion, for instance using the Alternating Direction Multiplier Methods (ADMM) [1]. These properties, allied to the advances in parallel computing of the past decades, have fostered applications of Augmented Lagrangian methods in a several disciplines. In particular, in control engineering, Augmented Lagrangian methods have been applied in discrete-time Model Predictive Control (MPC) [2] and discrete-time Nonlinear Model Predictive Control (NMPC) [3]. In these domains, the Augmented Lagrangian enabled the distributed solution the MPC and NMPC problems in discrete time.

Unlike in discrete-time control, the application of Augmented Lagrangian methods to solve Optimal Control Problems (OCP) in continuous-time system is much less developed. To this end, this paper contributes to the field of optimal control by proposing an Augmented Lagrangian method for optimal control of Differential-Algebraic Equations (DAE), accounting for constraints in states, algebraic, and control variables.

The approaches to solve the OCP can be divided in two big groups: the direct methods (those that find a solution that applying gradient decent methods to a objective function), and the indirect methods (those that find a solution for the optimality conditions). The Indirect Methods generally use a reduced number of variables, while direct methods discretize the system of equations and faces a size vs. accuracy dilemma. The Direct methods tends to have a faster convergence, in special for large scale systems, in which case the Indirect Methods might fail to converge. Another advantage of the Direct Methods is the ease to include constraint in the controls, states, and algebraic variables; while the Indirect Methods can only manage constraints on the control variables efficiently. In this work we were to suppress two of the weakness of the Indirect Methods, the inability to include constraints in the states and algebraic variables in practical way, and the problem with convergence.

The algorithm proposed in this paper obtain the solution of an OCP by solving a sequence of relaxed OCPs, which can be solved by any method, Direct or Indirect. However, in this paper, we chose the Indirect methods to demonstrate the advantages of such combination.

The main achievements of the proposed algorithms

- The reduction of a system of Algebraic Differential Equations (DAE) to a system of Ordinary Differential Equations (ODE), which from a implementation point of view makes the problem easier to treat since the latter has more mathematical and numerical tools.
- The OCP problems resultant from the application of this method tend to be easier to solve given that they are solved over a less restricted set.
- The method allows us to solve OCP with constraints in the states and algebraic variables, which is impractical with Indirect Methods.

II. Background

System dynamics can be modeled in different manners, *e.g.* using ordinary differential equations (ODE), partial differential equations (PDE), and differential algebraic equations (DAE). In this work, we are particularly interested in DAE systems, which is able to model a large class of systems. For optimal control of DAE systems, the optimality conditions are found in the literature and therefore they are briefly discussed herein. Because the algorithm to be developed reduces a DAE system into a ODE system, necessary optimality condition for optimal control of the latter are also presented.

A. Optimality Conditions for ODE Systems

An ODE system is one whose dynamics are generally given in the form

$$\dot{x} = f(x, u, t) \tag{1}$$

where $x(t) \in X$ is the state variable defined in the space $X = \mathbb{R}^{N_x}$, $u(t) \in U$ is the control variable defined in the space $U = \mathbb{R}^{N_u}$, and the time variable t is defined in the interval $[t_0, t_f]$. The function describing the dynamics of the state x is assumed to be continuously differentiable with respect to the variables x and u. The initial condition for the state variable is given by the vector x_0, such that $x(t_0) = x_0$.

An Optimal Control Problem (OCP) for an ODE system in the form (1) with a bounded control can be put in the form

$$\mathcal{P}_{ODE}: \quad \min \quad \Phi(x(t_f)) + \int_{t_0}^{t_f} L(x,u,t)\,dt \tag{2a}$$

$$\text{s.t.:} \quad \dot{x} = f(x,u,t) \qquad \forall t \in [t_0, t_f] \tag{2b}$$

$$x(0) = x_0 \tag{2c}$$

$$u(t) \in U_B \tag{2d}$$

The objective functional is defined by the dynamic cost function L and the final cost function Φ, which are both assumed to be continuously differentiable with respect to x and u. The set U_B is set defined by

$$U_B = \{\mathrm{u} \in U \,|\, u_L \leq \mathrm{u} \leq u_U\} \tag{3}$$

where u_L is the lower bound and u_U is the upper bound for the control actions. Without loss of generality, the variables t_0 and t_f are assumed to be fixed values, such that $t_0 < t_f$.

The necessary optimality conditions for the problem $\mathcal{P}_{ODE}$ are usually expressed with respect to the Hamiltonian function

$$H_{ODE}(x,\lambda,u,t) = L(x,u,t) + \lambda^T f(x,u,t) \tag{4}$$

where $\lambda(t) \in \mathbb{R}^{N_x}$ is the adjoint variable, also known as costate. From [4] the necessary conditions for (x^*, λ^*, u^*) to be optimal to $\mathcal{P}_{ODE}$ are

$$\dot{x}^* = \frac{\partial H_{ODE}}{\partial \lambda} = f(x^*, u^*, t) \tag{5a}$$

$$-\dot{\lambda}^* = \frac{\partial H_{ODE}}{\partial x} = \frac{\partial L}{\partial x} + \lambda^{*T} \frac{\partial f}{\partial x} \tag{5b}$$

$$u^*(t) = \arg\min_{\mathrm{u}\in U_B} H_{ODE}(x^*(t), \lambda^*(t), \mathrm{u}, t) \tag{5c}$$

$$\lambda^*(t_f) = \frac{\partial \Phi}{\partial x}(x^*(t_f), t_f), \qquad x^*(t_0) = x_0 \tag{5d}$$

which forms a system of $2N_x + N_u$ variables that are defined by $2N_x$ differential equations and N_u algebraic equations. For the differential equations, final conditions are imposed on the costates and initial conditions on the system states by (5d), respectively. Since a mix of initial and final conditions are given, the problem is characterized as a Boundary Value Problem (BVP) [5].

B. Optimality Conditions for DAE systems

A large class of systems can expressed as a semi-explicit DAE, which has the form

$$\dot{x} = f(x,y,u,t) \tag{6a}$$

$$0 = g(x,y,u,t) \tag{6b}$$

for $t \in [t_0, t_f]$ with $x(t_0) = x_0$, where $y(t) \in Y = \mathbb{R}^{N_y}$ is the algebraic variable, and the function g defines the algebraic variable, being assumed continuously differentiable with respect to x, y, and u. In addition, for a semi-explicit DAE as (6), the partial derivative $\frac{\partial g}{\partial y}$ is nonsingular for all $t \in [t_0, t_f]$ [5].

An OCP for the DAE in the form (6) with a bounded control can be stated in the form:

$$\mathcal{P}_{DAE}: \quad \min \quad \Phi(x(t_f)) + \int_{t_0}^{t_f} L(x,y,u,t)\,dt \tag{7a}$$

$$\text{s.t.:} \quad \dot{x} = f(x,y,u,t) \qquad \forall t \in [t_0, t_f] \tag{7b}$$

$$g(x,y,u,t) = 0 \qquad \forall t \in [t_0, t_f] \tag{7c}$$

$$x(0) = x_0 \tag{7d}$$

$$u \in U_B \tag{7e}$$

where $\nu(t) \in \mathbb{R}^{N_y}$ is the multiplier vector for the algebraic function g.

The Hamiltonian function for the OCP (7) can is given by

$$H_{DAE}(x,\lambda,y,\nu,u,t) = L(x,y,u,t) + \lambda^T f(x,y,u,t) + \nu^T g(x,y,u,t) \tag{8}$$

Using the Hamiltonian (8), the necessary conditions for $(x^*, \lambda^*, y^*, \nu^*, u^*)$ to be optimal to the OCP (7) are given by [6]:

$$\frac{\partial H_{DAE}}{\partial x} = -\dot{\lambda}^* = \frac{\partial L}{\partial x} + \lambda^{*T}\frac{\partial f}{\partial x} + \nu^{*T}\frac{\partial g}{\partial x} \tag{9a}$$

$$\frac{\partial H_{DAE}}{\partial y} = \frac{\partial L}{\partial y} + \lambda^{*T}\frac{\partial f}{\partial y} + \nu^{*T}\frac{\partial g}{\partial y} = 0 \tag{9b}$$

$$u^*(t) = \arg\min_{\mathrm{u}\in U_B} H_{DAE}(x^*(t), \lambda^*(t), y^*(t), \nu^*(t), \mathrm{u}, t) \tag{9c}$$

$$\frac{\partial H_{DAE}}{\partial \lambda} = \dot{x}^* = f(x^*, y^*, u^*), \quad \lambda^*(t_f) = \frac{\partial \Phi(x(t_f))}{\partial x(t_f)} \tag{9d}$$

$$\frac{\partial H_{DAE}}{\partial \nu} = g(x^*, y^*, u^*) = 0, \qquad x^*(0) = x_0 \tag{9e}$$

for all $t \in [t_0, t_f]$.

The system of equations given for necessary conditions (9) has $2N_x + 2N_y + N_u$ variables. The state is regulated by the differential equation that appears in (9d) with initial conditions established in (9e). Likewise, the costate is governed by the differential equation (9a), having its final state imposed by (9d). The control is induced by the minimization (9c), the algebraic multiplier is defined by (9b), and the algebraic variable is obtained from (9e).

Notice that the conditions (9) are only necessary for optimality of OCP (7). If the optimum control u^* lies in the interior of U_B for all t, then the minimization of u^* in (9c) satisfies the condition $\frac{\partial H}{\partial u} = 0$. However, there will be times when the optimal control will be at the boundary of U_B, in which case the condition $\frac{\partial H_{DAE}}{\partial u} = 0$ may not be verified.

Let us assume that the Hamiltonian H is convex with respect to u. If the solution of $\frac{\partial H}{\partial u} = 0$ induces a solution $\hat{u}(t)$ that lies in the interior of U_B, then $u^*(t) = \hat{u}(t)$ at time t. Otherwise, if $\hat{u}(t)$ does not lie in the interior of U_B, then, in such cases, the optimum is obtained by applying the value of the violated bounds [4]. For instance, if $\hat{u}(t) \in \mathbb{R}$ is above u_U at some time $t = t_1$, then $u^*(t_1) = u_U$. Therefore, given that $\hat{u}$ is the solution to $\frac{\partial H_{DAE}}{\partial u} = 0$, (9c) can be written for a convex Hamiltonian as:

$$u^*(t) = \begin{cases} u_U, & \text{if } u_U \leq \hat{u}, \\ \hat{u} & \text{if } u_L < \hat{u} < u_U, \\ u_L, & \text{if } \hat{u} \leq u_L \end{cases} \tag{10}$$

The same approach is valid for optimality conditions of the ODE case.

From a practical standpoint, when the convexity of the Hamiltonian cannot ascertained, then the procedure (10) can be seen as a best effort strategy.

C. Indirect Methods

The methods that solve the BVP consisting of the optimality conditions, equations (5) for ODEs and (9) for DAEs, are known as Indirect Methods. In contrast, Direct Methods minimize the objective functional (7a), while satisfying the constraints (7b)–(7e), in order to reach a solution. Although both classes of methods serve the same purpose, they achieve the solution in a different manner.

Each class can be split in implicit and explicit methods. The implicit methods make use of a black-box procedure to obtain a solution for underlying ODE/DAE system and its sensitivities [6]. On the other hand, the explicit approaches express the solution of the ODE/DAE system as a set of nonlinear equations. Shooting and collocation methods are arguably the most representative for implicit and explicit methods respectively.

1) Shooting Methods: Shooting methods solves a BVP through a sequence of Initial Value Problems (IVP). It works by solving an IVP (called a "shoot") for an initial guess, evaluating if the final boundary conditions are met, and making corrections for the guess in the next shoot. The shoot process can be represented, mathematically, by evaluating a function $F : \mathbb{R}^{N_x} \times T \rightarrow \mathbb{R}^{N_x}$ that, for an initial condition x_0, returns a final condition x_f of a given DAE system during time interval T. The boundary conditions are modeled by the function $G : \mathbb{R}^{N_x} \rightarrow \mathbb{R}^{N_x}$. The Single-Shooting Methods (SSM) consists in finding a vector x_0 that satisfies the system of nonlinear equations

$$x_f = F(x_0, [t_0, t_f]) \tag{11a}$$

$$0 = G(x_0, x_f) \tag{11b}$$

which can be solved by any nonlinear optimization algorithm, for instance Newton's method.

The SSM is not suitable for unstable BVPs [5], as those that arise from conditions (5) and (9) which invariably have unstable states or costate. From the duality nature of state-costate, if the state is stable, then the costate is unstable, and vice versa. For this reason, we choose the Multiple-Shooting Method for solving BVPs which is more robust to instability.

The Multiple-Shooting Method (MSM) breaks the integration interval in N subintervals. In each subinterval T_i, the function F is evaluated for an initial condition x_0^i to obtain the final state x_f^i of T_i. The initial state of T_i and the final state of T_{i-1} must agree to ensure state continuity. Further, the initial state of the first subinterval and the final state of last subinterval must satisfy the boundary conditions, given by function G. Altogether, they lead to the equations

$$x_f^i = F(x_0^i, T_i) \qquad i = 1, \dots, N \tag{12a}$$

$$x_f^{i-1} = x_0^i \qquad i = 2, \dots, N \tag{12b}$$

$$0 = G(x_0^1, x_f^N) \tag{12c}$$

which can be solved using the same tools of the SSM.

2) Collocation Method: Collocation is widely used method for solving OCPs using sparse nonlinear solvers [6]. Similar to MSM, it breaks the integration interval into N subintervals. However, instead of relying on a numerical integrator to solve the underlying ODE/DAE system, the Collocation Method approximates the trajectory of the states, algebraic variables, and controls by a parametrized function, in each subinterval. Typically, a Lagrangian interpolation polynomial $\widetilde{x}_i$ is used to approximate x within an interval T_i. The approximation ensures that the differential and algebraic equations are satisfied at some particular points, known as the collocation points.

In relation to MSM, Collocation consists in replacing the numerical integrator embedded in function F (12a) with a system of nonlinear equations. To ensure that the polynomial $\widetilde{x}_i$ approximates the trajectory of x, the derivative of $\widetilde{x}_i$ is forced to be equal to f at the collocation points.

III. Augmented Lagrangian for Constrained Optimization Problems

In mathematical programming, the Augmented Lagrangian Method is used to solve an equality Constrained Optimization Problem (COP) through a sequence of Unconstrained Optimization Problem (UOP). Let COP be of the form:

$$\min_z \quad V(z) \tag{13a}$$

$$\text{s.t.:} \quad c(z) = 0 \tag{13b}$$

The Augmented Lagrange method relax the equality constraint (13b) and includes a penalization term in the objective function creating an augmented objective function:

$$V_{\mu_k}(z, \lambda_k) = V(z) + \lambda^T c(z) + \frac{\mu}{2} \|c(z)\|^2 \tag{14}$$

where $\mu_k > 0$ is a scalar that belongs to sequence $\{\mu_k\} \rightarrow \infty$, and λ_k is approximation of the Lagrange multiplier of the constraint $c(z)$, which belongs to a sequence $\{\lambda_k\} \rightarrow \lambda^*$ [7].

The solution of (13) is obtained by a sequence of unconstrained minimizations of (14), determined by a scalar μ_k and a vector λ_k that are updated at each iteration. The method is outlined in Algorithm 1 [8].

A traditional rule for updating parameter μ_k, in line 7, is

$$\mu_{k+1} = \beta \mu_k \tag{15}$$

where β is a scalar greater than 1, usually in the range from 5 to 10. However, if μ_k is large, then the minimization of (14)

Algorithm 1 Augmented Lagrangian for Constrained Optimization

```
Require: $\mu_0 > 0$, $\varepsilon_{V,0} > 0$, starting points $z_0^s$ and $\lambda_0$:
for $k = 0, 1, \dots$ do
    Find a $z_k$ that minimizes $V_{\mu_k}(z, \lambda_k)$, starting at $z_k^s$, satisfying $\left\| \frac{\partial V_{\mu_k}}{\partial z}(z_k, \lambda_k,) \right\| \leq \varepsilon_{V,k}$,
    if $z_k$ satisfies a convergence condition, then
        return the solution $z_k$,
    end if
    Obtain $\lambda_{k+1}$ with the equation $\lambda_{k+1} = \lambda_k + \mu_k c(z_k)$,
    Choose a new parameter $\mu_{k+1} \geq \mu_k$,
    Set the starting point for the next iteration $z_{k+1}^s = z_k$,
    Select tolerance $\varepsilon_{V,k+1}$
end for
```

might become ill conditioned [7]. To this end, an alternative update rule is

$$\mu_{k+1} = \begin{cases} \beta\mu_k & \text{if } \beta\mu_k < \mu_{\max} \\ \mu_{\max} & \text{otherwise} \end{cases} \tag{16}$$

There exists a theoretical value μ^* that, for any $\mu > \mu^*$, $\{V_\mu(z_k, \lambda_k)\} \to V(z^*)$ where z^* is a solution for (13), if the tolerance $\varepsilon_{V,k+1} \to 0$ as $k \to \infty$ and the problem satisfies some conditions [8].

IV. Algorithm

For the constrained optimization problem (13), the standard Augmented Lagrangian Method relaxes the equality (13b) and then solves a sequence on unconstrained problem. By analogy, the algebraic equations of the optimal control problem of a DAE system (7) could be relaxed according with the Augmented Lagrangian. This section presents an Augmented Lagrangian algorithm to solve OCPs with algebraic equations, following the structure developed for constrained optimization, briefly described in Section III.

Given a optimal control problem in the form $\mathcal{P}_{DAE}$, the algorithm relaxes the algebraic equation (7c) which is then penalized in the objective, introducing the new functional

$$J_\mu(x, y, u, \nu) = \Phi(x(t_f)) + \int_{t_0}^{t_f} \mathcal{L}_\mu(x, y, u, \nu, t)\, dt \tag{17}$$

where the function $\mathcal{L}_\mu$ is defined by

$$\begin{aligned} \mathcal{L}_\mu(x, y, u, \nu, t) = F(x, y, u, t) + \nu(t)^T g(x, y, u, t) \\ + \frac{\mu}{2} \left\| g(x, y, u, t) \right\|^2, \end{aligned} \tag{18}$$

where the $\mu > 0$ is a scalar and the *function* $\nu : [t_0, t_f] \to \mathbb{R}^{N_y}$ is an approximation for the multiplier function ν^* that satisfies the optimality conditions (9) for the OCP $\mathcal{P}_{DAE}$.

The functional (17) is the objective of an auxiliary optimal control problem solved by the algorithm at each iteration k, which is given by

$$\mathcal{P}_{\mathcal{L}}(\mu_k, \nu_k):$$

$$\min_{y,u} J_{\mu_k} = \Phi(x(t_f)) + \int_{t_0}^{t_f} \mathcal{L}_{\mu_k}(x, y, u, \nu_k, t)\, dt \tag{19a}$$

$$\text{s.t.: } \dot{x} = f(x, y, u, t), \quad \forall t \in [t_0, t_f] \tag{19b}$$

$$x(0) = x_0 \tag{19c}$$

$$u(t) \in U_B \quad \forall t \in [t_0, t_f] \tag{19d}$$

Notice that without an algebraic equation, the variable y is free to be optimized. In this sense, the algebraic variable plays the same role as the control variable u. Therefore, we define an extended control variable $\widehat{u} = [u, y]$, where $\widehat{u} \in \widehat{U} = U_B \times Y$. Using $\widehat{u}$, the problem $\mathcal{P}_{\mathcal{L}}$ meets the standard form of an OCP with ODE (2), and the optimality conditions (5) apply.

The Augmented Lagrange Method for Optimal Control is stated in Algorithm 2. Therein, the parameter μ_0 is the initial value of the sequence $\{\mu_k\}$, ν_0 is the initial function for the sequence $\{\nu_k\}$, and ε_g is the tolerance on the violation of the algebraic constraint g.

Algorithm 2 Augmented Lagrangian for Optimal Control

```
Require: $\mu_0$, $\nu_0$, $K$, $\varepsilon_J$, and $\varepsilon_g$:
1: $J_0 \leftarrow \infty$
2: for $k = 1, 2, \dots$ do
3:     $(J_k, x_k, y_k, u_k) \leftarrow solve\{\mathcal{P}_{\mathcal{L}}(\mu_k, \nu_k)\}$
4:     $\nu_{k+1} \leftarrow \nu_k + \mu_k g(x_k, y_k, u_k)$
5:     $\mu_{k+1} \leftarrow update\_mu\{\mu_k\}$
6:     if $\|g(x_k, y_k, u_k)\| < \varepsilon_g$, $\forall t \in [t_0, t_f]$ then
7:         return $u_k$
8:     end if
9: end for
```

Some particularities of this algorithm deserve discussion:

1) In line 3, the pseudo-function *solve* produces a solution to the OCP $\mathcal{P}_{\mathcal{L}}$ using any suitable method, direct or indirect. In the case of y_k being parametrized, for instance as a polynomial in direct method, then y_k should be a sufficiently good approximation in order to not hinder convergence. To speed up the algorithm, the solution of the previous iteration can be used as an initial guess at the current iteration.
2) Line 4 updates the multiplier ν_k. Because generic functions cannot be stored in computers, ν_k is approximated with a parametric function which, in this work, is the same Lagrange interpolation polynomial used in the Collocation Method. Any educated guess for ν_0 should used, otherwise by defining $\nu_0 = 0$ the algorithm will consider only the quadratic penalty at the first iteration.
3) In line 5, the pseudo-function *update_mu* increments the penalty parameter μ_k. The traditional rule for Augmented Lagrange is (15), however the alternative rule (16) can be applied to prevent ill conditioning.

To a great extent, the contributions of this work are the following desirable properties of the algorithm:

1) By relaxing the algebraic equations, the algorithm transforms the DAE system into an ODE system. This reduction renders optimal control more applicable, given that ODE solvers are readily available and have a reduced computational cost.
2) The algorithm solves an OCP of the form (7) which accounts for bound constraints on control variables. In addition, it can cope with bound constraints on algebraic algebraic and state variables, that is, $y(t) \in Y_B$ and $x(t) \in X_B, \forall t \in [t_0, t_f]$, with

$$Y_B = \{\mathrm{u} \in Y \,|\, y_L \leq \mathrm{y} \leq y_U\} \tag{20a}$$

$$X_B = \{\mathrm{u} \in X \,|\, x_L \leq \mathrm{x} \leq x_U\} \tag{20b}$$

where y_L and y_U are bounds for the algebraic variables, and x_L and x_U are bounds for the state variables.

By noticing that $\widehat{u} = [u, y]$, the second property can be shown by defining $\widehat{u}(t) \in \widehat{U}_B = U_B \times Y_B$ and using the condition (5c) to obtain a solution to $\mathcal{P}_{\mathcal{L}}$ that, by consequence, ensures $y(t) \in Y_B$.

To satisfy $x(t) \in X_B$, a variable $y_x \in X_B$ is introduced in the original OCP (7) along with the algebraic equation

$$y_x = x \tag{21}$$

Then, by defining an extended control variable $\widehat{u} = [u, y, y_x] \in \widehat{U}_{BX} = U_B \times Y_B \times X_B$, the solution of the problem $\mathcal{P}_{\mathcal{L}}$ using the (5c) implies $x(t) \in X_B$.

V. Experiments

This section presents results from computational experiments that substantiate the distinct properties of the Augmented Lagrangian Method.

A. Experimental Setup

The analysis consider the Van der Pol oscillator [9], which has an unstable equilibrium at 0 and an attractive limit cycle. These features render the oscillator a widely used benchmark for control of nonlinear systems. The Van der oscillator can be modeled in the form of a ODE system by

$$\dot{x}_1 = (1 - x_2^2)x_1 - x_2 + u \tag{22a}$$

$$\dot{x}_2 = x_1 \tag{22b}$$

For the purpose of our analysis, the ODE system (22) is recast as a DAE system,

$$\dot{x}_1 = y + u \tag{23a}$$

$$\dot{x}_2 = x_1 \tag{23b}$$

$$y = (1 - x_2^2)x_1 - x_2 \tag{23c}$$

With the objective of keeping the system at the unstable equilibrium, the following optimal control objective was chosen

$$J(x, y, u) = \int_{t_0}^{t_f} \left[x_1^2 + x_2^2 + u^2\right] \, dt \tag{24}$$

Let us define the optimal control problem $\mathcal{P}_{ODE}^{V}$ as the one that minimizes the functional J (24) and subject to the ODE system (22). In addition, let the optimal control problem $\mathcal{P}_{DAE}^{V}$ consist of the minimization of the functional J (24), while being subject to the DAE system (23).

To investigate the properties of the proposed algorithm, three cases were considered for the optimal control problems:

- Case 1, compares the solutions of $\mathcal{P}_{ODE}^{V}$ and $\mathcal{P}_{DAE}^{V}$ obtained by solving BVPs, with the solution of $\mathcal{P}_{DAE}^{V}$ using the proposed algorithm.
- Case 2, considers the same settings of Case 1, however the control variable is constrained by $-0.3 \leq u \leq 1$.
- Case 3, solves only $\mathcal{P}_{DAE}^{V}$ subject to the bound constraints $-0.3 \leq u \leq 1$ on the control variables and the constraint $-0.4 \leq x_1$ on the state x_1. Only the proposed algorithm can handle this problem.

Case 1 and Case 2 intend to show that all the approaches induces the same optimal values. Case 3 highlights the algorithm's ability of handling bounds on state variables.

For all the test cases, the Multiple Shooting and Collocation Method were used to solve the BVPs resulting from the optimality conditions.

B. Boundary Value Problem

For Case 1, we explicitly define the BVP arising from the optimality conditions for the relaxed OCP of the form (19), which is iteratively solved by Algorithm 2 to obtain a solution to $\mathcal{P}_{DAE}^{V}$. First, the algebraic equation (23c) is relaxed and penalized in the objective functional. By defining $g(x, y, u) = (1 - x_2^2)x_1 - x_2 - y$, the functional becomes

$$J_\mu(x, y, u, \nu) = \int_{t_0}^{t_f} \Bigg[x_1^2 + x_2^2 + u^2 + \nu^T g(x, y, u) + \frac{\mu}{2}\left\|g(x, y, u)\right\|^2\Bigg] \, dt \tag{25}$$

The relaxed OCP for Case 1 minimizes the functional (25) subject to the ODE system formed by (23a) and (23b). Given that $\hat{u} = [u, y] \in X = U \times Y$, the optimality conditions for OCP of ODEs (5) yields the BVP of the DAE:

$$\dot{x}_1^* = y^* + u^*, \qquad \dot{x}_2^* = x_1^* \tag{26a}$$

$$-\dot{\lambda}_1^* = (2x_1) + [\nu + \mu g(x^*, y^*, u^*)](1 - x_2^{*2}) + \lambda_2 \tag{26b}$$

$$-\dot{\lambda}_2^* = (2x_2^*) + [\nu + \mu g(x^*, y^*, u^*)](-2x_1^* x_2^* - 1) \tag{26c}$$

$$0 = 2u^* + \lambda_1, \tag{26d}$$

$$0 = \nu(-1) + \mu g(x^*, y^*, u^*)(-1) + \lambda_1^* \tag{26e}$$

$$x(t_0) = x_0, \qquad \lambda_1(t_f) = \lambda_2(t_f) = 0 \tag{26f}$$

in view that (26c) and (26c) are algebraic equations.

From (26e), the optimal controls can be derived

$$u^* = -\frac{\lambda_1^*}{2}, \qquad y^* = -\frac{\nu}{\mu} - (1 - x_2^2)x_1^* - x_2^* + \frac{\lambda_1^*}{\mu} \tag{27}$$

By substituting (27) for u^* and y^* in (26), the system (26) becomes a BVP of an ODE.

C. Experiments Results

The experiments where implemented in the automatic differentiation framework CasADi (version 3.0.0) [10] which uses the numerical integrator Sundials CVODES and IDAS to solve the ODE and DAE systems, CasADi makes use of nonlinear solver IPOPT [11] to solve the nonlinear system of equations. The parameters used for the Augmented Lagragian algorithm was $\beta = 8$, $\mu_0 = 2$, $\nu_0 = 0$, and $\varepsilon_g = 10^{-6}$

The analysis of the results will be done with respect to the convergence of the objective, the convergence of the violation of the algebraic constraint, and the feasibility of the x_1.

1) For the Case 1, the ODE, DAE and Augmented Lagrangian resulted in the same objective value of 2.86695, with the Multiple-Shooting and the Collocation Method. Case 2 had the same outcome, all of the tests resulted in the objective value 2.87972. The solution of Case 3 resulted in the objective 2.95321, which agrees with the results reported in [12]. Table I display the time taken to obtain the solution in each approach.
2) To measure the violation, we used the root mean square (RMS) of the the algebraic function g, which is given by

$$RMS(g) = \sqrt{\int_{t_0}^{t_f} \|g(x_k, y_k, u_k, t)\|_2^2 \, dt} \qquad (28)$$

Figure 1 shows the RMS of the algebraic function g in each iteration of the algorithm while solving the OCP of Case 1, using both, the Collocation and Multiple-Shooting Methods. The figure shows that the algorithm presented a superlinear convergence for the experiment, taking only 6 iterations to reduce the violation to very small values. It is worth to say that after solving the first iteration the subsequent solutions are fast obtained.
3) Figure 2 shows the optimal control and the evolution of the states obtained for for Case 3 using the Multiple Shooting Method. The control profile has the same shape than the control profiles depicted in [12], for the same problem.

Table I
SOLUTION TIME OBTAINED WITH THE EXPERIMENTS

		ODE	DAE	Aug. Lagrangian
Case 1	Coll.	0.69 s	0.89 s	0.95 s
	MSM	0.45 s	0.83 s	1.75 s
Case 2	Coll.	0.64 s	0.86 s	0.97 s
	MSM	0.72 s	1.32 s	1.88 s
Case 4	Coll.			1.22 s
	MSM			3.18 s

VI. Conclusion

In this work we proposed and algorithm that is able to obtain the solution of OCP of DAE systems through a sequence of OCP of ODE system. This approaches allows the use of ODE solvers, which have reduced computational cost and are more available. Another property of the transformation is the ease of including bound constraints on the states, which otherwise, would not be possible using the common approaches of Indirect Methods. The results of the experiments have shown that the proposed method have competitive solution time with the respect to the traditional DAE approach, while ensure a sufficiently small violation in the algebraic constraint, using less specialized tool, and solving a larger class of problems.

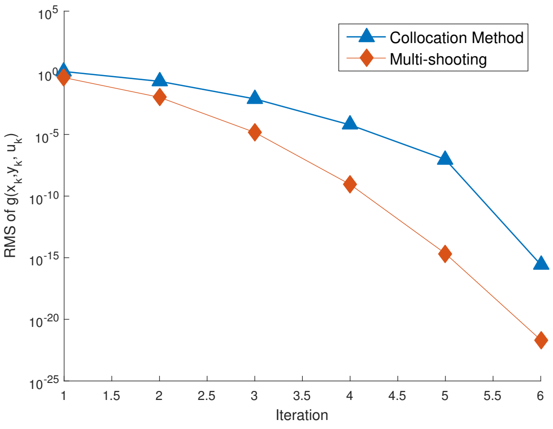

Figure 1. RMS of $g(x_k, y_k, u_k, t)$ for Case 1

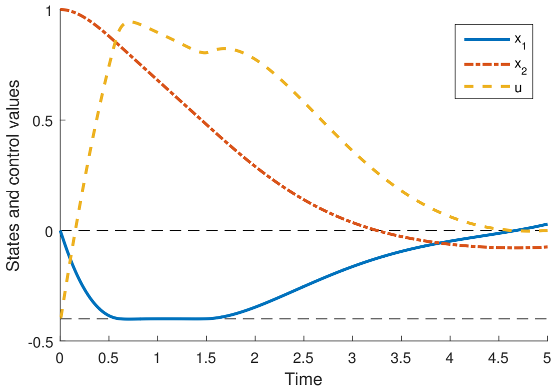

Figure 2. Plot of the optimal trajectories for Case 1

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, 2011.

[2] M. J. Köegel and R. Findeisen, "Cooperative distributed MPC using the alternating direction multiplier method," in *8th IFAC Advanced Control of Chemical Processes*, 2012.

[3] S. Hentzelt and K. Graichen, "An augmented lagrangian method in distributed dynamic optimization based on approximate neighbor dynamics," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2013.

[4] D. Kirk, *Optimal Control Theory: An Introduction*, ser. Dover Books on Electrical Engineering Series. Dover Publications, 2004.

[5] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, jan 1998.

[6] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Process*. Society for Industrial and Applied Mathematics, 2010.

[7] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[8] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.

[9] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[10] J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," PhD thesis, Arenberg Doctoral School, KU Leuven, 2013.

[11] A. Wächter and T. L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2005.

[12] M. Bell and R. Sargent, "Optimal control of inequality constrained DAE systems," *Computers & Chemical Engineering*, 2000.