Computational reduction of optimal hybrid vehicle energy management

Armenta, Carlos; Delprat, Sebastien; Negenborn, Rudy R.; Haseltalab, Ali; Lauber, Jimmy; Dambrine, Michel

DOI
10.1109/LCSYS.2020.3046609

Publication date
2021

Document Version
Accepted author manuscript

Published in
IEEE Control Systems Letters

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2020.3046609, IEEE Control Systems Letters

Computational reduction of optimal hybrid vehicle energy management

Carlos Armenta, Sébastien Delprat, Rudy R. Negenborn, Ali Haseltalab, Jimmy Lauber, and Michel Dambrine

***Abstract*— Pontryagin's Minimum Principle is a way of solving hybrid powertrain optimal energy management. This paper presents an improvement of a classical implementation. The core of this improvement consists in relaxing the tolerance on some intermediate steps of the algorithm in order to reduce the number of iterations and thereby reducing the number of operations required to compute an optimal solution. The paper describes both a classical implementation of Pontryagin's Minimum Principle as well as the improved version. Numerical simulations are conducted on an academic example to demonstrate the benefits of the proposed approach.**

***Index Terms*— Optimal control, control applications, numerical algorithms.**

I. Introduction

HYBRID powertrains use at least two energy sources for their propelling, and at least one of them is reversible. A control strategy is needed to manage the power-split between the different sources such that a criterion, for instance, the fuel consumption is minimized.

In simulation over a priori known mission, the problem is to compute an optimal solution to the energy management problem. It can be obtained, for instance, using Dynamic Programming. It consists of reformulating the energy management problem as a shortest path problem within an oriented graph with positive costs [1] [2] [3]. It can handle both state and control constraints seamlessly. Still, it is subject to the so-called 'curse of dimensionality' that restricts the algorithm to problems with a single energy storage system. Another approach is based on Pontryagin's Minimum Principle (PMP) [4] [5]. This approach provides the necessary conditions that allow computing the optimal solution. The resulting algorithm is more computationally efficient than DP, but it cannot handle state constraints.

Real-time control algorithms can be derived from optimal conditions. They are known as Equivalent Consumption Minimization Algorithms [6]. Considering potential real-time applications, for instance, within the Model Predictive Control framework, the reduced computational cost of the PMP based algorithms is of importance. Improving optimal control algorithm efficiency is also of interest for topology an sizing optimization. It consists in choosing the powertrain components and also the way they should be connected [7] [8]. This is typically done by solving many optimization problems for different numerical settings (such as component power rating, energy storage capacity) over a large set of mission profiles. This procedure can be computational intensive since it relies on an almost exhaustive search. As a result, improving the optimal control algorithm efficiency allows either expanding the search space or reducing the computation time.

Problem statement: The classical implementation of the PMP-based algorithm consists of two major optimization sub-problems. First, at each instant, the control is a solution to an instantaneous optimization sub-problem that depends on an additional parameter denoted as co-state and second, the co-state is the root of a function which is computed using a bisection search.

Contribution: The algorithm is modified to improve the computational efficiency of the algorithm significantly. The underlying idea is that both sub-problems do not need to be solved with high accuracy before their convergence toward the optimal solution. As a result, the intermediate solutions are computed using weak but sufficient accuracy by exploiting the convexity properties of both subproblems.

Organization: The results of this work are presented as follows: Sect. II briefly introduces the theoretical background on PMP and how it can be used to obtain a numerical solution to a hybrid powertrain optimal energy management problem; Sect. III details how to improve the classical approach; Sect. IV puts the proposed improved control scheme and the classical one at test in an academic example in simulation; conclusions are given in Sect. V.

Manuscript received September 14, 2020; revised October 24, 2020; accepted December 8, 2020. This work has been supported by the ISHY Interreg 2 seas project through the PhD Position "Energy management of hybrid ships".

C. Armenta, S. Delprat, J. Lauber, and M. Dambrine are with LAMIH UMR CNRS 820, University Polytechnic Hauts-de-France, F-59313 Valenciennes, France. (e-mail: Sebastien.Delprat@uphf.fr; Jimmy.Lauber@uphf.fr; Michel.Dambrine@uphf.fr; Carlos-daniel.Armentamoreno@uphf.fr).

R. Negenborn and A. Haseltalab are with the Department of Marine and Transport Technology, Delft University of Technology, Delft, Netherlands (e-mail: R.R.Negenborn@tudelft.nl; A.Haseltalab@tudelft.nl).

II. Preliminaries

In this paper, a series hybrid vessel is considered. In order to focus on the proposed improvement, a simplified hybrid powertrain model is considered with perfect energy storage and a perfect DC/DC converter. However, the presented algorithm can be extended to other hybrid powertrain topologies with more detailed models.

A. Series Hybrid topology

In the series hybrid topology, the vessel is exclusively propelled by the electric motor, see Fig. 1. The propelling power is supplied to this traction motor by the energy storage system (typically a battery) and/or the Auxiliary Power Unit (APU).

The latest can be a fuel cell system [9] or, as in this paper, an ICE coupled to a generator. The power to be delivered to the load (i.e. traction motor) is denoted by w, and it is subject to the following power split equation:

$$w(t) = y(t) + u(t), \tag{1}$$

with $y(t) \in \left[\underline{y}, \overline{y}\right]$ being the battery power and $u(t) \in [0, u_{\max}]$ the APU output power. In order to lighten the expressions, the

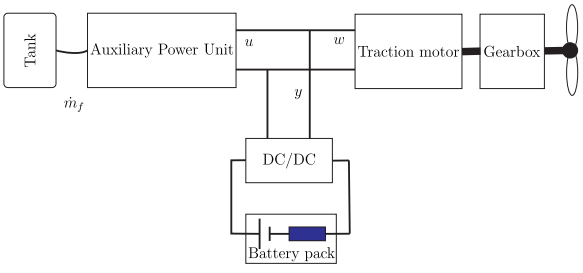

Fig. 1. Powertrain diagram

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2020.3046609, IEEE Control Systems Letters

dependence on the time variable t is omitted when there is no ambiguity. The battery state of charge dynamics is:

$$\dot{x}(t) = f(u, w) = \frac{-y(t)}{Q} = \frac{u(t) - w(t)}{Q}, \tag{2}$$

where Q is the storage capacity and x the battery state of charge. The APU fuel consumption to be minimized is given as:

$$\min_{u(\cdot)} J[u] = \int_0^T \dot{m}_f(u(\tau))d\tau, \tag{3}$$

with the fuel mass flow rate $\dot{m}_f$ assumed to be convex in u. Considering eq. (1) and both the APU output power and DC/DC limitations, the set of admissible inputs is defined as $\mathcal{U}(w) = [\underline{u}, \overline{u}]$ with $\underline{u} = \max(0, -\overline{y} + w(t))$ and $\overline{u} = \min(u_{\max}, -\underline{y} + w(t))$.

B. Pontryaging's Minimum Principle

In simulation, the required power profile $w(t)$ to be supplied to the load is known over the optimization horizon $[0, T]$. It can be measured on an existing vessel or computed using a velocity profile and a model [10]. The vessel energy management can be formulated as an optimal control problem:

$$\min_{u(\cdot)} J[u] = \int_0^T \dot{m}_f(u(\tau))d\tau \tag{4}$$

subject to

$$\dot{x} = f(u, w), \tag{5}$$

$$\mathcal{U}(w) = [\underline{u}, \overline{u}], \tag{6}$$

$$x(0) = x_0 \text{ and } x(T) = x_T, \tag{7}$$

with x_0 and x_T the initial and final state of charge.

C. Necessary conditions for optimality

Let us define the Hamiltonian function as:

$$H(u, \lambda, w) = \dot{m}_f(u) + \lambda^T f(u, w), \tag{8}$$

with λ being the co-state. PMP establishes a necessary condition for optimality [5]:

$$\frac{\partial H}{\partial \lambda} = \dot{x}(t) \tag{9}$$

$$\frac{\partial H}{\partial x} = -\dot{\lambda}(t) \tag{10}$$

$$u(t) = \arg\min_{\nu \in \mathcal{U}} H(\nu, \lambda(t), w(t)). \tag{11}$$

According to (2) and (10), the co-state is constant:

$$\dot{\lambda}(t) = 0 \Rightarrow \lambda(t) = \lambda_0, \quad \forall t \in [0, T], \tag{12}$$

λ_0 being a constant to be determined. Furthermore, it is assumed that the Hamiltonian is convex in the control. As a result, PMP optimality conditions are also sufficient. From (11), let us denote the optimal policy by Π, a function depending on the load power $w(t)$ and the co-state value λ_0:

$$u(t) = \Pi(\lambda_0, w(t)) = \arg\min_{\nu \in \mathcal{U}} H(\nu, \lambda_0, w(t)). \tag{13}$$

Thus, the original optimal control problem is reduced to a Boundary Value Problem (BVP) parametrized by a single unknown λ_0:

$$\dot{x}(t) = f(\Pi(\lambda_0, w(t)), w(t)), \tag{14}$$

$$x(0) = x_0, \quad x(T) = x_T. \tag{15}$$

Considering any arbitrary value for λ_0, the initial state of charge x_0 being known, the final state of charge value $x(T)$ can be determined by direct integration:

$$x(T) = x(0) + \int_0^T f(\Pi(\lambda_0, w(t)), w(t))dt. \tag{16}$$

Finally, the initial costate λ_0 is the root of the following defect function:

$$g(\lambda_0) = \tilde{x}_T(\lambda_0) - x_T. \tag{17}$$

In order to solve the optimal control problem, one must solve two sub-problems: (i) the Hamiltonian minimization (11) and (ii) the co-state computation (17).

In order to prove that it is possible to find λ_0 such that $g(\lambda_0) = 0$, let us introduce the following property

Theorem 2.1: Let $\lambda_0 \leq 0$, $-\partial \dot{m}_f / \partial u$ be a strictly decreasing function with respect to u, and f defined in (2), then the optimal policy Π is a decreasing function with respect to λ_0 and there is a monotonic increasing relation between the value of λ_0 and the final state of charge $x(T)$.

Proof: Let us first consider the u_c the unconstrained solution to the Hamiltonian minimization:

$$u_c = \arg\min_{\nu} H(\nu, \lambda_0, w(t)). \tag{18}$$

H has a local minimum if $\partial H / \partial u = 0$, then

$$\frac{\partial H}{\partial u}(u_c, \lambda_0, w) = \frac{\partial \dot{m}_f}{\partial u}(u_c) + \frac{1}{Q}\lambda_0 = 0 \Rightarrow \lambda_0 = -Q\frac{\partial \dot{m}_f}{\partial u}(u_c), \tag{19}$$

since $-\partial \dot{m}_f / \partial u\,(u_c)$ is a strictly decreasing function with respect to u_c, the relation between the optimal control u_c and λ_0 is strictly decreasing.

The Hamiltonian being convex in u, and considering the control saturation such that $u \in \mathcal{U}(w)$, the optimal policy (solution to the constrained Hamiltonian minimization (13)) is:

$$u = \min(\overline{u}, \max(\underline{u}, u_c)). \tag{20}$$

As a result, the optimal control policy Π is a monotonic decreasing function of λ_0. The second assertion follows from the latter and Lemma 1 in [11]. ■

Theorem 2.1, implies that $g(\lambda_0)$ is only a monotonic function of λ_0 (and not a strictly monotonic one). Due to the control saturation $u \in \mathcal{U}(w)$, the reachable final state of charge set for $x(T)$ is bounded. As a result, g has a null derivative for small and large λ_0 values that may induces numerical issues when solving (21) using root finding algorithms such as Newton's method. Instead, derivative-free algorithms are preferred and the bisection method is considered. The co-state is obtained by numerically solving the following optimization:

$$\lambda_0 = \arg\min_{\lambda \in [\underline{\lambda}, \overline{\lambda}]} |g(\lambda)|, \tag{21}$$

with g being monotonic, there always exist sufficiently small (resp. large) $\underline{\lambda}$ (resp. $\overline{\lambda}$) such that $g(\underline{\lambda}) > 0 > g(\overline{\lambda})$.

In order to numerically estimate the integral in eq. (16), the Euler numerical quadrature is used:

$$x(T) \approx x(0) + \sum_{i=0}^{N-1} f(\Pi(\lambda_0, w(i \cdot s)), w(i \cdot s))s, \tag{22}$$

with s as the sampling period and $T = (N-1) \cdot s$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2020.3046609, IEEE Control Systems Letters

D. Classical algorithm implementation

To solve the Hamiltonian minimization (11), the control is gridded and the Hamiltonian is evaluated at each node. The grid size is denoted by M and at every instant $i = 0, 1, \ldots, N$, the control grid is $U_{grid}(i)$. Let us denote by $u_{grid}(i,k)$ the elements of $U_{grid}(i)$ sorted from the smallest to the largest such that $u_{grid}(i,k) < u_{grid}(i,k{+}1)$, $\forall k = 0, \ldots, M$. The considered grid for the classical algorithm is defined as follows:

$$u_{grid}(i,k) = \underline{\nu}(i) + k\left(\overline{\nu}(i) - \underline{\nu}(i)\right)/M, \tag{23}$$

$\forall i = 0, \ldots, N$ and $\forall k = 0, \ldots, M$ with $\overline{\nu}(i) = \overline{u}$, $\underline{\nu}(i) = \underline{u}$. As a result, the optimal policy (13) is replaced by the following approximation:

$$\Pi_{grid}\left(\lambda_0, w(i\cdot s)\right) = \underset{\nu_k \in U_{grid}(i)}{\arg\min}\ H\left(\nu_k, \lambda_0, w(i\cdot s)\right), \tag{24}$$

with $U_{grid}(i)$ defined as follows

$$U_{grid}(i) = \{u_{grid}(i,k), \forall k = 0, \ldots, M\}. \tag{25}$$

This way, for a given λ_0 and $w(i\cdot s)$, the optimal control is estimated with an accuracy ϵ_N:

$$\left|\Pi_{grid}\left(\lambda_0, w\left(i\cdot s\right)\right) - \Pi\left(\lambda_0, w\left(i\cdot s\right)\right)\right| < \epsilon_N, \tag{26}$$

with $\epsilon_N = \frac{\overline{u}-\underline{u}}{2N}$.

The final state of charge is estimated Euler quadrature:

$$\tilde{x}(T) = x(0) + \sum_{i=0}^{N-1} f(\Pi_{grid}(\lambda_0, w(i\cdot s)), w(i\cdot s))s. \tag{27}$$

The co-state is obtained by computing the roots of the following defect function using a bisection search:

$$\tilde{g}(\lambda_0) = \tilde{x}_T(\lambda_0) - x_T. \tag{28}$$

Algorithm 1 shows the detailed procedure. Although being quite simple and widespread [12] [13] [14] [15], this algorithm can be improved. First, let us note that to ensure a good accuracy on the optimal policy the grid U_{grid} is defined with a constant number of vertices N large enough. During the initial iterations of the bisection algorithm, the co-state value is far away from the final one. As a result, all these iterations are carried out with a fine grid even if a high accuracy on the control is not needed at that stage. In this paper, we propose to adapt ϵ_N according to the accuracy required by the bisection algorithm. As a result, the overall number of Hamiltonian evaluations is reduced.

Algorithm 1 Classical algorithm

1: Set $\underline{\lambda}$, $\overline{\lambda}$ and the inputs $w(t)$, x_T, N_T, $u_{\max}$, $\underline{y}$, $\overline{y}$
2: Define U_{grid}
3: **do**
4: $\quad\lambda_0 = (\overline{\lambda} + \underline{\lambda})/2$
5: $\quad$**for** $i = 0$ to $N-1$ **do**
6: $\quad\quad\Pi_{grid}(i) = \underset{\nu_k \in U_{grid}}{\arg\min}\ H(\nu_k, \lambda_0, w(i\cdot s))$
7: $\quad$**end for**
8: $\quad$Compute $\tilde{x}(T)$ using (27)
9: $\quad$**if** $\tilde{x}(T) > x_T$ **then**
10: $\quad\quad\overline{\lambda} = \lambda_0$
11: $\quad$**else**
12: $\quad\quad\underline{\lambda} = \lambda_0$
13: $\quad$**end if**
14: **while** $|\tilde{x}(T) - x_T| > \Delta$

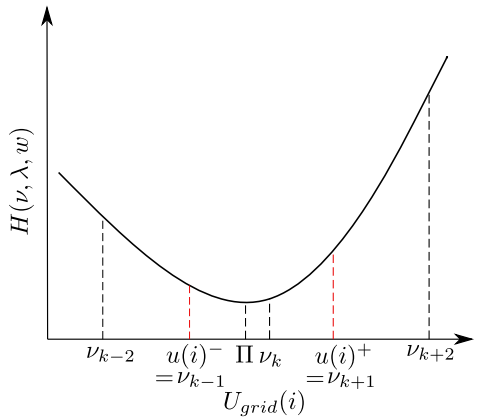

Fig. 2. Tracking of the optimal control

III. Improved algorithm implementation

As the Hamiltonian minimum is not computed exactly but estimated using the grid U_{grid}, one should carefully track the actual value of the true optimal control policy Π with respect to the grid vertices. As a result, the improved algorithm consists in reducing the grid U_{grid} to the smallest required set and adapting it at each iteration. At every instant $i = 0, 1, \ldots, N$, $U_{grid}(i)$ is a vector whose entries are the elements from the set (25) sorted from the smallest to the largest. Thus, it is possible to define the numerical estimate of the optimal control Π_{grid} and the control brackets u^-, and u^+ as follows:

$$k = \underset{\nu_k \in U_{grid}(i)}{\arg\min}\ H(\nu_k, \lambda_0, w(i\cdot s)), \tag{29}$$

$$\Pi_{grid}(i) = \nu_k, \tag{30}$$

$$u^-(i) = \nu_{k-1}, \tag{31}$$

$$u^+(i) = \nu_{k+1}. \tag{32}$$

As depicted in Fig. 2, the actual control input that minimises the Hamiltonian, namely Π, is always bracketed by the interval $[v_{k-1}, v_{k+1}]$ due to the Hamiltonian convexity. The grid can be refined using a smaller step size ϵ_N and then, at the next iteration the difference $v_{k+1} - v_{k-1}$ will be reduced, and so the approximation of the optimal solution will be closer to its actual value. The final state of charge $x(T)$ obtained using the optimal control (22) is also bracketed by $\tilde{x}_{u^-}$ and $\tilde{x}_{u^+}$, with :

$$\tilde{x}_{u^-} = x(0) + \sum_{i=0}^{N-1} f(u^-(i), w(i\cdot s))s, \quad \tilde{x}_{u^+} = x(0) + \sum_{i=0}^{N-1} f(u^+(i), w(i\cdot s))s. \tag{33}$$

In order to implement the bisection algorithm, it is necessary to determine if the final state $x(T)$ is greater or lower than x_T (step 9 of Algorithm 1). Exploiting Theorem 1, three cases may occur:

- $\tilde{x}_{u^-}, \tilde{x}_{u^+} > x_T$, then the optimal final state for this co-state exceed the target $\tilde{x}(T) > x_T$. At each instant i, the optimal control $u(i)$ is lower than $u^+(i)$. Any value above $u^+(i)$ can be removed from $U_{grid}(i)$:

$$U_{grid}(i) \leftarrow U_{grid}(i) \cap \left\{\nu \in U_{grid}(i) : \nu \leq u^+(i)\right\} \quad \forall i = 0, \ldots, N. \tag{34}$$

- $\tilde{x}_{u^-}, \tilde{x}_{u^+} < x_T$, then the optimal final state for this co-state subceed the target $\tilde{x}(T) < x_T$. At each instant i, the optimal control $u(i)$ is greater than $u^-(i)$. Any value below $u^-(i)$ can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2020.3046609, IEEE Control Systems Letters

be removed from $U_{grid}(i)$:

$$U_{grid}(i) \leftarrow U_{grid}(i) \cap \left\{ \nu \in U_{grid}(i) : \nu \geq u^-(i) \right\} \quad \forall i = 0, \ldots, N. \tag{35}$$

- $\tilde{x}_{u^-} < x_T < \tilde{x}_{u^+}$, then the location of the actual $\tilde{x}(T)$ with respect to x_T is unknown. It is therefore necessary to refine the grid according to Theorem 3.1 (step 10) by adding two extra values and removing $\Pi_{grid}(i)$ from the grid:

$$U_{grid}(i) \leftarrow U_{grid}(i) \cup \left[\frac{2u^-(i)+u^+(i)}{3}, \frac{u^-(i)+2u^+(i)}{3} \right] - \{\Pi_{grid}(i)\}, \forall i = 0, \ldots, N. \tag{36}$$

The intuitive idea is to add gridding points closer and closer to the optimal value. Also, by this method, at the beginning of the algorithm, U_{grid} might contain a small number of points to be evaluated in (11). Algorithm 2 shows the detailed procedure.

Algorithm 2 Improved algorithm

```
Set λ̲, λ̄ and the inputs w(t), x_T, N, u_max, y̲, ȳ
do
    Initialize λ_0 = (λ̄ + λ̲)/2
    for i = 0 to N − 1 do
        k(i) = argmin_{ν_k ∈ U_grid(i)} H(ν_k, λ_0, w(i·s))
        u^-(i) = ν_{k−1}
        u^+(i) = ν_{k+1}
    end for
    Compute x̃_{u^-} and x̃_{u^+} according to (33).
    if (x_T − x̃_{u^-})(x_T − x̃_{u^+}) > 0 then
        if x̃_{u^-} > x_T then
            λ̲ = λ_0
            Remove values in U_grid according to (34)
        else
            λ̄ = λ_0
            Remove values in U_grid according to (35)
        end if
    else
        for i = 0 to N do
            Refine U_grid according to (36)
        end for
    end if
while |x_T − x̃_{u^-}| > Δ || |x_T − x̃_{u^+}| > Δ
```

Theorem 3.1: Let $H_j(u)$, with $j = 1, 2, \ldots, p$ be a sequence of convex functions on a convex domain $\mathcal{D}$, and assume that the sequence converges to a function $H(u)$. Then $H(u)$ is convex.

Proof: Assume by contradiction that we have a pair of points u_1, u_2, and $0 < \alpha < 1$ such that, defining $u = \alpha u_1 + (1-\alpha)u_2$, we have

$$H(u) > \alpha H(u_1) + (1-\alpha)H(u_2). \tag{37}$$

On the other hand, from the convexity assumption $-H_j(u) + \alpha H_j(u_1) + (1-\alpha)H_j(u_2) \geq 0$, $\forall j$. Taking the limit, we have $H(u) \leq \alpha H(u_1) + (1-\alpha)H(u_2)$, which contradicts (37). ∎

Consider the sequence $\{H_n(u)\}$ of Hamiltonian convex functions, such that the value of the co-state λ_0 determines every element of the sequence. Thus, if λ_0 converges using the bisection algorithm then, from Theorem 3.1, the sequence of Hamiltonian functions converges to a convex function $H(u)$.

Each time the grid is refined according to (36) a new pair of optimal control brackets u^- and u^+ will be included within it for the next iteration. Then, two sequences $\{u_n^-\}$,$\{u_n^+\}$ are defined, where the subscript n stands for the number of refinements in the grided set. For a specific n in the sequence, two cases might occur:

1) $u_n^- = u_{n-1}^-$ and $u_n^+ = \dfrac{u_{n-1}^- + 2u_{n-1}^+}{3}$.
2) $u_n^- = \dfrac{2u_{n-1}^- + u_{n-1}^+}{3}$ and $u_n^+ = u_{n-1}^+$.

The convergence of the improved algorithm toward the optimal solution is then guaranteed by the Theorem 3.2.

Theorem 3.2: Let $\{u_n^-\}, \{u_n^+\}$ be a pair of bounded sequences in the interval $[a, b]$, with $u_1^- < u_1^+$ and defined by the following rule

$$u_n^- = \begin{cases} u_{n-1}^- & H\left(\frac{2u_{n-1}^- + u_{n-1}^+}{3}\right) > H\left(\frac{u_{n-1}^- + 2u_{n-1}^+}{3}\right) \\ \frac{2u_{n-1}^- + u_{n-1}^+}{3} & \text{otherwise} \end{cases}$$

$$u_n^+ = \begin{cases} u_{n-1}^+ & H\left(\frac{u_{n-1}^- + 2u_{n-1}^+}{3}\right) > H\left(\frac{2u_{n-1}^- + u_{n-1}^+}{3}\right) \\ \frac{u_{n-1}^- + 2u_{n-1}^+}{3} & \text{otherwise} \end{cases}$$

for any $n > 2$ and any convex function H. Then the sequences $\{H(u_n^-)\}, \{H(u_n^+)\}$ converge, moreover, they converge to the same limit.

Proof: Suppose that $H\left(\frac{2u_{n-1}^- + u_{n-1}^+}{3}\right) < H\left(\frac{u_{n-1}^- + 2u_{n-1}^+}{3}\right)$ holds, then

$$H(u_n^-) = H(u_{n-1}^-) \text{ and } H(u_n^+) = H\left(\frac{u_{n-1}^- + 2u_{n-1}^+}{3}\right).$$

Considering that

$$H\left(\frac{2u_{n-1}^- + u_{n-1}^+}{3}\right) < H(u_n^+) \iff H(u_n^+) < 2H(u_n^+) - H\left(\frac{2u_{n-1}^- + u_{n-1}^+}{3}\right).$$

Since H is convex the following holds:

$$H\left(u_n^+\right) \leq 2H\left(u_n^+\right) - H\left(\frac{2u_{n-1}^- + u_{n-1}^+}{3}\right) \leq \frac{2}{3}H\left(u_{n-1}^-\right) + \frac{4}{3}H\left(u_{n-1}^+\right) - \left(\frac{2}{3}H\left(u_{n-1}^-\right) + \frac{1}{3}H\left(u_{n-1}^+\right)\right) = H\left(u_{n-1}^+\right).$$

Thus, $H\left(u_n^+\right) \leq H\left(u_{n-1}^+\right)$. The proof is analogous in the other case. Since the sequences $\{H(u_n^-)\}$ and $\{H(u_n^+)\}$ are monotonically decreasing and bounded, then they converge to a minimum and, H being convex, they converge to its global minimum. ∎

IV. Example

In this section, the classical and the improved algorithms are applied to a CTV vessel in operation over two days. In order to focus on the algorithm performances, a simplified model is considered. The relation between the number of Hamiltonian evaluations required by the classical algorithm against the improved one for different values of accuracy is presented for comparison.

The load $w(t)$, is computed using a model that relates the propeller power as a function of the ship speed profile, see Fig. 3, and the vessel characteristics [16]. The load value is $w(t) = 2\pi n_p Q_p$; where n_p is the propeller speed, and Q_p is the propeller torque. For the sake of simplicity, a squared resistance-speed function is considered, and it is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCSYS.2020.3046609, IEEE Control Systems Letters

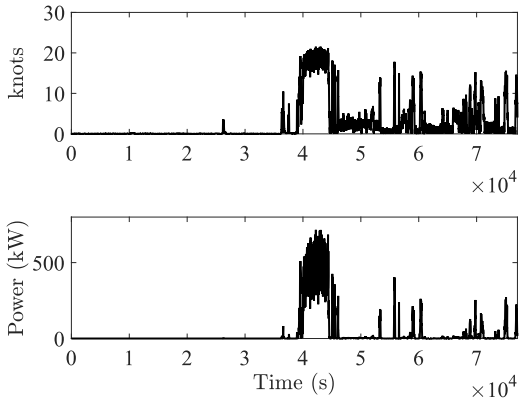

Fig. 3. Ship speed profile (above) and the effective power profile (below)

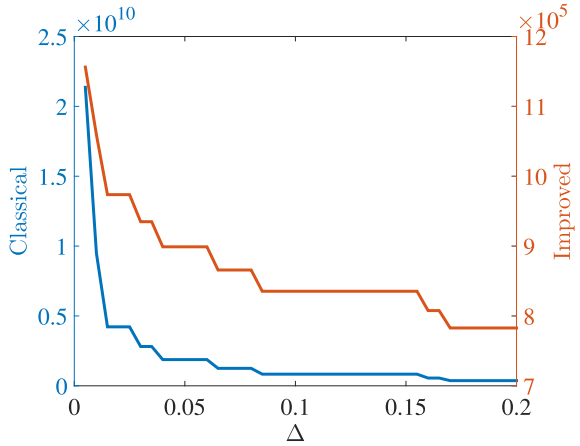

Fig. 4. Number of Hamiltonian evaluation: classical approach vs improved approach.

assumed that the wake factor is constant. Moreover, the relationship between the shaft speed and propeller speed is established as follows:

$$n_p = \frac{1 - f_w}{JD} v_s, \tag{38}$$

where f_w is the wake factor, D is the propeller diameter, J is the advance ratio, and v_s is the vessel speed. Considering that the ship uses a fixed-pitch propeller with a constant operation point, then the advance ratio J remains constant, and the propeller speed is proportional to the speed of the ship. Then, the propeller torque is established using the equation

$$Q_p = K_Q \rho n_p^2 D^5, \tag{39}$$

where K_Q is the torque coefficient, which can be expressed as a polynomial in terms of J and since J is a constant, K_Q is a constant.

The vessel and energy storage system parameters are presented in Table I. The bisection algorithm parameters are $\underline{\lambda} = -500$ and $\overline{\lambda} = 0$, with the boundary conditions $x_0 = x_T = 0.5$. The simulation time step size for the ship model is $s = 56s$ while $N = 1382$. To make a fair comparison of both algorithms, the solutions must be computed with the same accuracy ϵ_N. So, first, for a given final state of charge accuracy Δ, the improved algorithm is executed, and the final optimal control accuracy ϵ_N is obtained. Then the classical bisection algorithm is run using ϵ_N as a control grid accuracy in (24). 40 values of the final state of charge tolerance Δ linearly spaced within the interval $[0.005\ 0.2]$ are considered. The accuracy achieved by the improved algorithm ranges from $\epsilon_N = 42.2$ (kW) down to $\epsilon_N = 0.7$ (kW), depending on the tolerance value. Fig. 4 shows the number of Hamiltonian evaluations that were performed with both algorithms for different tolerance values. The computed costate for the optimal solution that matches the boundary condition with a tolerance $\Delta = 0.5\%$ is $\lambda = -2.573$ for both algorithms. Let us define the improvement factor $\mu(\epsilon_N)$ as the ratio of Hamiltonian evaluations between the classical and improved algorithm. The effectiveness of the improved algorithm is illustrated in Fig. 5, showing that even for a very sparse control grid, generated using $\epsilon_N = 42.2\,kW$, the improvement factor is greater than 473. For a very refined grid the improvement factor can reach values greater than four orders of magnitude. It is important to notice that as long as ϵ_N tends to 0 the ratio tends to grow exponentially, making it clear that the more accurate the solution is required, the more efficient the improved algorithm is. The optimal solution which matches the boundary condition with a tolerance $\Delta = 0.5\%$ with $\lambda = -2.573$, the running time for the classical algorithm is 827.4459 seconds, whereas, for our proposal it is 9.4249 seconds.

Symbol	Description	Value	Units
D	Propeller Diameter	0.5	m
ρ	Density of sea water	1024	kg/m^2
J	Advance ratio	0.73	-
f_w	Wake factor	0.19	-
K_Q	Propeller torque coefficient	0.0199	-
Q	Battery capacity	150	CAh

TABLE I
MODEL PARAMETERS

V. CONCLUSION

An improvement of the classical implementation of an algorithm used for solving hybrid powertrain optimal energy management has been presented. It has been shown that the number of computations of the Hamiltonian is significantly reduced using the proposed algorithm, even in the cases where the required accuracy is low. The methodological improvement has been demonstrated over a simplified series hybrid case and a recorded mission profile. Future work will be devoted to the extension to more complex vessel architectures encountered in maritime applications.

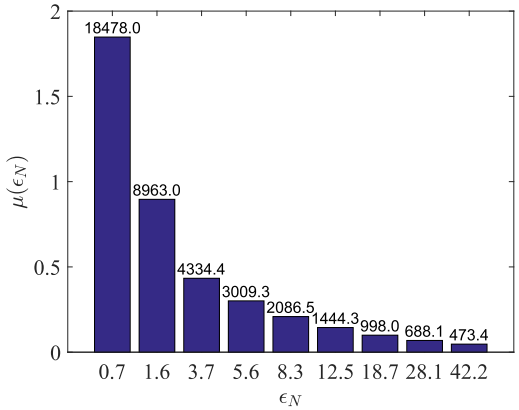

Fig. 5. Relation between the number of operations of the classical and the improved approach

REFERENCES

[1] Z. Wei, J. Xu, and D. Halim, "Hev energy management fuzzy logic control based on dynamic programming," in *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*. IEEE, 2015, pp. 1–5.

[2] J. Lempert, B. Vadala, K. Arshad-Aliy, J. Roeleveld, and A. Emadi, "Practical considerations for the implementation of dynamic programming for hev powertrains," in *2018 IEEE Transportation Electrification Conference and Expo (ITEC)*. IEEE, 2018, pp. 755–760.

[3] S. Uebel, N. Murgovski, C. Tempelhahn, and B. Bäker, "Optimal energy management and velocity control of hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 327–337, 2017.

[4] S. Delprat, J. Lauber, T. M. Guerra, and J. Rimaux, "Control of a parallel hybrid powertrain: optimal control," *IEEE transactions on Vehicular Technology*, vol. 53, no. 3, pp. 872–881, 2004.

[5] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2004.

[6] S. Yang, J. Wang, F. Zhang, and J. Xi, "Self-adaptive equivalent consumption minimization strategy for hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, 2020.

[7] T. Hofman, S. Ebbesen, and L. Guzzella, "Topology optimization for hybrid electric vehicles with automated transmissions," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2442–2451, 2012.

[8] E. Silvas, T. Hofman, N. Murgovski, L. P. Etman, and M. Steinbuch, "Review of optimization strategies for system-level design in hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 57–70, 2016.

[9] K. Ettihir, M. H. Cano, L. Boulon, and K. Agbossou, "Design of an adaptive ems for fuel cell vehicles," *International Journal of Hydrogen Energy*, vol. 42, no. 2, pp. 1481–1489, 2017.

[10] R. Geertsma, R. Negenborn, K. Visser, M. Loonstijn, and J. Hopman, "Pitch control for ships with diesel mechanical and hybrid propulsion: Modelling, validation and performance quantification," *Applied Energy*, vol. 206, pp. 1609–1631, 2017.

[11] T. van Keulen, J. Gillot, B. de Jager, and M. Steinbuch, "Solution for state constrained optimal control problems applied to power split control for hybrid vehicles," *Automatica*, vol. 50, no. 1, pp. 187 – 192, 2014.

[12] N. Robuschi, M. Salazar, P. Duhr, F. Braghin, and C. H. Onder, "Minimum-fuel engine on/off control for the energy management of a hybrid electric vehicle via iterative linear programming," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 134–140, 2019.

[13] C. Hou, M. Ouyang, L. Xu, and H. Wang, "Approximate pontryagin's minimum principle applied to the energy management of plug-in hybrid electric vehicles," *Applied Energy*, vol. 115, pp. 174–189, 2014.

[14] S. Delprat, T. Hofman, and S. Paganelli, "Hybrid vehicle energy management: Singular optimal control," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 9654–9666, 2017.

[15] L. V. Pérez, C. H. De Angelo, and V. Pereyra, "Determination of the adjoint state evolution for the efficient operation of a hybrid electric vehicle," *Mathematical and Computer Modelling*, vol. 57, no. 9-10, pp. 2257–2266, 2013.

[16] H. K. Woud and D. Stapersma, *Design of propulsion and electric power generation system*. IMAREST, 2002.