Energy Optimal Real-Time Navigation System

Tomas Jurik
ESIEE Paris, France and Brno University of Technology, Czech Republic
E-mail: hello@tomasjurik.com

Arben Cela
UPE, ESIEE Paris, 93162, Noisy-Le-Grand Cedex, France, and Laboratoire des Signaux et Systèmes, CNRS/Supelec, 91192, Gif-Sur-Yvettes Cedex, France, E-mail: a.cela@esiee.fr

Redha Hamouche, René Natowicz, and Abdellatif Reama
UPE, ESIEE Paris, 93162, Noisy-Le-Grand Cedex, France
E-mails: r.hamouche@esiee.fr, r.natowicz@esiee.fr, a.reama@esiee.fr

Silviu-Iulian Niculescu
Laboratoire des Signaux et Systèmes, CNRS/Supelec, 91192, Gif-Sur-Yvettes Cedex, France
E-mail: silviu.niculescu@lss.supelec.fr

Jérôme Julien
AKKA Technologies, 41 Boulevard Vauban, 78280 Guyancourt, France
E-mail: jerome.julien@akka.eu

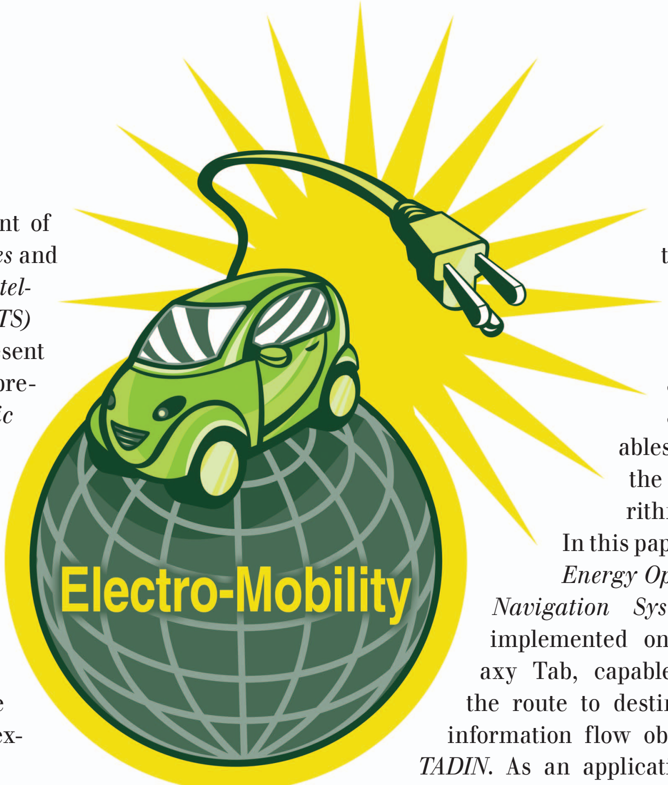

IMAGE LICENSED BY INGRAM PUBLISHING

***Abstract*—**The rapid development of *Mobile Internet* and *Smart Devices* and advent of a new generation of *Intelligent Transportation Systems (ITS)* increase information about present driving conditions and make its prediction possible. *Real time traffic information systems (TIS)* like *SYTADIN* help in route to destination planning and traffic state prediction. Energy-optimal routing for electric vehicles creates novel algorithmic challenges where the computation complexity and the quality of information on traffic state are the main issues. This complexity is induced by the possible negative values of edge energy as well as the variability of route and vehicle variables which render the standard algorithms unsuitable. In this paper we present an *Energy Optimal Real Time Navigation System (EORTNS)*, implemented on Samsung Galaxy Tab, capable of calculating the route to destination based on information flow obtained from *SYTADIN*. As an application example we propose a real time energy management for a Hybrid Electrical Vehicle (*HEV*) composed

Digital Object Identifier 10.1109/MITS.2014.2324000
Date of publication: 23 July 2014

1939-1390/14©2014IEEE

of batteries and Super-Capacitors (SC). The *EORTNS* is not only capable of energy optimal route to destination calculation with respect to traffic state but also operates the *On-Board* power splitting between batteries and Super-Capacitors.

The HEV on-board energy sources efficiency is increased substantially by optimally splitting the demanded power between them.

I. Introduction

The convergence of communication and sensing on multimedia platforms such as smart devices provides the engineering community with unprecedented monitoring capabilities. They include a video camera, numerous sensors (accelerometers, light sensors, *GPS*, microphone), wireless communication outlets (*GSM*, *GPRS*, Wi-Fi, Bluetooth, infrared), computational power and memory. Due to their portability, computation, and communication capabilities, smart devices are transforming our cars into moving sensors capable of communicating their position and operating real time decision[1], [2]. The concomitant advent of mobile Internet and smart devices promotes *Advance Driver Assistance Systems(ADASs)* application capable to address and resolve in real time the optimal vehicle navigation problem where traffic network state vary with time.

The Hybrid Electrical Vehicle (*HEV*) have the capability to regenerate the braking/deceleration energy which increases their range. Moreover, the *HEV* on-board energy sources efficiency is increased substantially by optimally splitting the demanded power between them [12]. The optimal power splitting can be operated in real time based on calculated origin to destination (*O-D*) 3D routes which may be time optimal, distance optimal or energy optimal. Whereas the time and distance metric give positive route segment costs the energy optimal metric can give negative ones. If for time optimal and distance optimal route calculation the routing algorithms like *Dijkstra* and its variants give generally good results, it is not the case for real time energy optimal metric one. The reasons are different. *First*, as aforementioned, the energy cost of route segment can be negative due to the 3*D* route profile (downhill moving) and brake regeneration. In order to apply *Dijkstra* algorithm and its variants, which can operate routing calculation only for graphs with positive edge weight, it is necessary to operate Johnson like pre-processing techniques [7] which are generally very time consuming. *Second*, because of traffic state and vehicle parameter variation, the energy cost calculation may be operated only in run-time and exclude the pre-processing techniques as *Johnson* [7] like algorithms based on global graph analysis in order to eliminate the negative energy costs. *Third*, the recuperation of the kinetic and potential energy, even for *HEV*, depends on the route choice as well as on the characteristics of the power sources such as their capacities and power efficiency and limits [10], [8], [9], [15], [16]. This implies the fact that the route energy cost is not necessarily equal to the sum of all segments costs composing this path.

The main contributions of this paper are summarized in the following points:

- We propose a real time solution of time constrained energy optimal navigation problem using the real-time traffic information flow provided by *SYTADIN* [11], a public *traffic information system (TIS).*
- The proposed speed-up techniques given in fourth section operate a search space reduction of routing algorithm helping to meet real-time constraints.
- We show the effectiveness of the proposed *EORTNS* system in real situations corresponding to energy optimal navigation in the *Paris region* which represents a directed graph with 207275 vertices and 467423 edges.

This paper is organized as follows. In the second section a review of the related research works is given. Problem formulation of energy optimal real time navigation problem as well as the method of road segment cost rescaling based on *Johnson* [7] method revisited in the context of electrical vehicle are given in section three. Fourth section is consecrated to time constrained energy optimal problem formulation and solution. A pre-processing algorithm which reduces substantially the searching space or number of nodes that may belong to the optimal path is given. An algorithm based on *path dominance* and *look-ahead* techniques [18], [17] is proposed to calculate the optimal path. In section five we give the general software architecture of *EORTNS*. An application example as well as *EORTNS* software implementation, its experimental results and their analysis are presented in sixth section. Section seven is consecrated to conclusions and future works.

II. Related Work

Energy optimal routing has recently attracted the interest of many researchers [10], [8], [9], [15], [16]. In *Sachenbacher* et al. [10] a solution of the energy optimal routing problem with battery capacity limitation within a A^* search framework is given. Consistent modeling of battery capacity constraints and recuperation energy are given as well as tests results demonstrating the efficiency of their algorithm of worst case complexity $O(nlogn + m)$, where n is the number of nodes and m is the number of edges of

We consider an energy optimal time constrained problem allowing us to render more realistic and acceptable the calculated route to destination.

the traffic network graph. Employing a generalization of *Johnson* potential technique, an energy optimal routing of an electrical vehicle is given in Eisner et al. [8] considering the on board energy capacity limitations as well as energy recuperation. Under non-constant edge costs they propose a routing algorithm of complexity $O(nlogn+m)$ increased by pre-processing complexity $O(nm)$. Wang et al. proposed in [15] a context aware energy optimization route for electrical cars. In their framework Wang et al. consider a permanent access to the real-time traffic network data and road segment energy cost calculation is operated in real time. A energy constrained time optimal $A^*(CA^*)$ algorithm (*ECO*) as well as a bi-objective criteria optimization algorithm (*LCO*) are proposed. The two problem are resolved and simulation results are given.

All these research works share the same concern that of road segment energy cost modeling with respect to road profile and energy recuperation. All of them consider constant speed in each road segments and the energy cost rescaling are equivalent and, in our case, their values remain constant. In [15] the road segment energy cost is considered dependent on the vehicle time entry in a given road segment which is a coarse approximation of the traffic flow and does not change fundamentally the problem. Unlike [10], [8], [9] we consider an energy optimal time constrained problem allowing us to render more realistic and acceptable the calculated route to destination. The driver acceptance to spend more time to reach the destination helps to decrease the energy used as well as creates the first conditions of a more elaborated routing algorithm based on a collaborative approach.

III. Problem Formulation

The road network graph is composed of vertices representing road junctions or intersection points and edges representing the road segment connecting them. The edges cost represents the real time energy necessary to transport the *HEV* along the road segment calculated at the query time. So the road network may be given by a directed graph $G=(V,E,c)$ where V represent the set of graph vertex or the intersection points of cardinality n and E represents the set of graph edges of cost $c(v_i,v_j):(V\times V)\Rightarrow R$ and cardinality m. We assume also that for each vertex the elevation function $z(v_i):V\Longrightarrow R_0^+$ is given. We suppose also that we know the road segment length $l(v_i,v_j):(V\times V)\Longrightarrow R^+$ and the real time road segment speed $s(v_i,v_j):(V\times V)\Longrightarrow R_0^+$. The road segments velocities are given by *SYTADIN* and are time varying with respect to the traffic state.

The optimal route or path $P_{v_o\to v_N}$ from origin point $\mathbf{s}$ to destination point $\mathbf{t}$ is given by an ordered sequence of vertices $(v_0,v_1,v_2,\ldots,v_k,v_N)$, where v_0 and v_N correspond to origin point $\mathbf{s}$ and destination point $\mathbf{t}$ respectively. Each sub-sequence of two consecutive elements of this ordered sequence is a road segment or an edge of the directed graph G. Note by $P_{v_o\to v_i}$ the sub-path from the origin to the point or vertex v_k. Note also by $\mathcal{P}$ the set of feasible routes from the origin to the destination point. It is clear that a given route to destination is feasible if all its sub-paths are feasible.

The graph edges cost represents the energy needed to move the vehicle from the initial road segment point to its final one. This energy depends on the vehicle characteristics, velocity, chosen route and its $3D$ profile. This means that their values have to be calculated only on-line and *Johnson* pre-processing technique [3] can not be applied. For the clarity of presentation we will give in the sequel quite briefly the main components composing the vehicle power and energy demand.

A. Longitudinal Model

The vehicular model of longitudinal dynamics is constructed based on the theory of vehicle multi-body dynamics [3], [5].

1) Vehicle Power Balance Equation

According to an analysis of the summation of performing forces acting on the vehicle body in the longitudinal direction, the power balance for the controlled vehicle is governed by:

$$\begin{aligned}P_D(t) &= f_{P_D}(v,\dot{v},M,\theta) = (F_w+F_r+F_c+F_a v)\\ &= \underbrace{\frac{\rho}{2}\cdot A_f\cdot C_d v^3}_{F_w v} + \underbrace{M\cdot g\cdot v\cdot(\mu_r\cdot\cos\theta+\sin\theta)}_{(F_r+F_c)v} +\\ &\quad + \underbrace{M\cdot(1+\delta_{\mathrm{eqm}})\cdot\dot{v}v}_{F_a v},\end{aligned} \tag{1}$$

where:

- F_w, F_r, F_c, F_a are respectively the forces corresponding to wind resistance, rolling resistance, gravitation and acceleration resistance
- $v, r, A_f, C_d, q, \delta_{\mathrm{eqm}}$ are respectively the vehicle speed, air density, vehicle frontal area, Reynolds coefficient, road slope and equivalent moment of inertia of vehicle rotating parts.

2) Position-Based Power Calculation

Road slope and vehicle speed are relevant with respect to optimal routing problem if they are dependent on position rather than time.

As an important parameter in equation (1), the road slope may change frequently at the actual road environment, especially in the mountain terrain, and has a major impact on the energy consumption. Road slope and vehicle speed are relevant with respect to optimal routing problem if they are dependent on position rather than time. Therefore, the time-based equation (1) is not suitable for optimal route to destination calculation and a power expression depended on position, s, is preferable. Using the transformation:

$$v = \frac{ds}{dt}$$
$$\frac{dv}{ds} = \frac{dt}{ds} \cdot \frac{dv}{dt} = \frac{1}{v} \cdot \frac{dv}{dt}, v \neq 0$$

the equation (1) can be transferred into a position-based form

$$P_D(s) = \frac{\rho}{2} A_f C_d \cdot v^3 + M \cdot g \cdot \mu_r \cdot v \cdot \cos\theta + M \cdot g \cdot v \cdot \sin\theta + \\ + M \cdot (1 + \delta_{\mathrm{eqm}}) \cdot v^2 \cdot \frac{dv}{ds}. \quad (2)$$

From the equation (2) and under the assumption that the velocity and the route slope are constant on each road segment([1]) we may give the energy needed to move the vehicle from the beginning to the end of the road segment (v_i, v_{i+1}) as:

$$E_D(i) = \frac{\rho}{2} \cdot A_f \cdot C_d \cdot v_i^2 \cdot l_i + M \cdot g \cdot \mu_r \cdot l_i \cdot \cos\theta_i \\ + M \cdot g \cdot l_i \cdot \sin\theta_i. \quad (3)$$

In the equation (3) the only term which may have negative value is the third one which corresponds to potential energy. If its absolute value is also superior to the sum of two other terms then the edge energy cost is negative. Lest us note by E_R the energy corresponding to wind and rolling resistance and by E_P the potential energy. So for each road segment i we can decompose the corresponding energy as follow:

$$E_R(i) = \frac{\rho}{2} \cdot A_f \cdot C_d \cdot v_i^2 \cdot l_i + M \cdot g \cdot \mu_r \cdot l_i \cdot \cos\theta_i \\ E_P(i) = M \cdot g \cdot l_i \cdot \sin\theta_i, \quad (4)$$

where $i \in \{0, \ldots, N-1\}$.

B. Road Segment Energy Cost Rescaling

The optimal route to destination is necessarily a feasible one and it optimizes the criteria. It means that battery and Super-Capacitors have sufficient energy to move the vehicle through each road segment. It is clear that each sub-path of the optimal one has to be feasible and its concatenation with each road segments originating from its last vertex remain feasible. The question we may ask is how to decide, in run time, which sub-sequence is feasible and which is not one and which modification on the energy model of road segment costs we have to operate. Intuitively, if the remaining energy is not sufficient to move the vehicle to the next vertex of the road segment its cost has to be infinity.

As aforementioned, we also have to rescale all the road segments cost in order to obtain only positive values. These necessary road segments energy cost model modifications have to be consistent with the optimal energy problem and helps to reduce the calculation complexity.

From (3) and (4), the road segment energy cost may be given by:

$$c_e(i) = E_R(i) + E_P(i). \quad (5)$$

In the case where only a partial downhill potential energy recuperation is possible we can rewrite the expression of segment energy costs (5) as:

$$c_e(i) = \begin{cases} E_R(i) + \alpha \cdot E_P(i) & \text{if } E_P(i) \leq 0 \\ E_R(i) + E_P(i) & \text{if } E_P(i) > 0 \end{cases}, \quad (6)$$

where α is the downhill potential energy recuperation coefficient $(\mathbf{0} < \alpha < \mathbf{1})$.

The road segments energy costs given by (5) and (6) may have negative values. In order to apply *Dijskra* like algorithms we have to transform them consistently in order to obtain positive road segment energy costs. For this we can use the conservative property([2]) of potential energy which help us to transform the relation (6) to:

$$c_e^m(i) = \begin{cases} E_R(i) + (\alpha - 1) \cdot E_P(i) & \text{if } E_P(i) \leq 0 \\ E_R(i) & \text{if } E_P(i) > 0 \end{cases}. \quad (7)$$

Let us define the sub-sequence $P^j_{v_0^j \to v_{i+1}^j} = (v_0^j, v_1^j, \ldots, v_{i+1}^j)$ of order i of the route to destination sequence $P^j_{v_0^j \to v_N^j} = (v_0^j,$

[1]The flow speed given by *SYTADIN* is constant. In order to better approximate the road slope we generate intermediary discretization point between two road segment vertices.

[2]In case of perfect recuperation of downhill potential energy, the route to destination does not depend on the altitude of intermediary points but only on those of origin and destination ones.

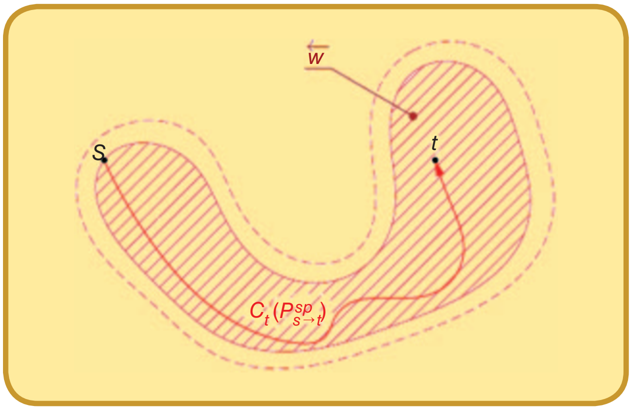

FIG 1 Example search space of the computation when the shortest path is found and this search is stopped. Nodes within the hatched area have the head path computed in the $\overleftarrow{w}$ set. The dashed line marks the area of nodes that can be reached within the constraint.

$v_1^j,\dots,v_N^j$) The cost of the sub-path $P^j_{v_0^j\to v_{i+1}^j}$ which is the rescaled sum of spent and recuperated energy is given by:

$$c(P^j_{v_0^j\to v_{i+1}^j}) = \sum_{k=0}^{k=i} c_e^m(k). \tag{8}$$

In order that expression (8) be consistent with routing problem, it has to integrate battery and super capacitor capacity constraint as well as theirs initial state of charge. In the same time we have also to consider the cases where there is no battery and/or super capacitor free capacity to recuperate downhill potential energy. As we shall see in the following, integrating the battery and super capacitors capacity constraints and theirs initial values operate modifications on the path cost function as well as on the road segment energy cost. For clarity of presentation we will suppose that the downhill energy may be stored to battery or supper capacitors. The case where we allow only to super-capacitors to store the downhill energy may be integrated easily.

Let us note respectively by $C_{\max}$ and C_{init} the maximum and initial state of charge of on-board energy representing the aggregated values of battery and super capacitors. Let us note also by:

$$\Delta^i = C(P^j_{v_0^j\to v_i^j}) + c_e(i), \quad (i = 1,\dots,N), \tag{9}$$

where $C(P^j_{v_0^j\to v_i^j})$ is given by [10]:

$$C(P^j_{v_0^j\to v_i^j}) = \begin{cases} C_{\max} - C_{\text{init}} & \text{if } i = 0 \\ 0 & \text{if } i > 0, \Delta^i < 0 \\ \Delta^i & \text{if } i > 0, 0 \le \Delta^i \le C_{\max} \\ \infty & \text{if } i > 0, \Delta^i > C_{\max} \end{cases}. \tag{10}$$

The cost function $C(P^j_{v_0^j\to v_i^j})$ in (10) gives the free on board storage energy capacity value which minimization is consistent with the minimum path energy cost. If the free on board energy capacity becomes equal to zero we lose partially or totally the recuperation capacity of road segment downhill potential energy which has to be considered in the model of the road segment energy cost. As proposed in [8], we define the additional road segment energy cost:

$$\hat{E}_D(i) = \begin{cases} -\Delta^i & \text{if } i > 0, \Delta^i < 0 \\ 0 & \text{if } i > 0, 0 \le \Delta^i \le C_{\max} \\ \infty & \text{if } i > 0, \Delta^i > C_{\max} \end{cases}. \tag{11}$$

So the total road segment energy cost which has to be considered for the routing, integrating the additional energy cost (11) is given by:

$$\begin{aligned} c_e^m(i) &= E_R(i) + (\alpha - 1)E_P(i) \\ &\quad + \hat{E}_D(i) \Leftarrow (E_P(i) \le 0) \wedge (\Delta^i < 0) \\ c_e^m(i) &= E_R(i) + \hat{E}_D(i) \Leftarrow (E_P(i) \ge 0) \wedge (\Delta^i \ge 0). \end{aligned} \tag{12}$$

The **energy optimal routing problem** can be formulate as [8]: *Given an energy graph $G = (V,E,c_e^m)$, the origin and destination points $s,t \in V$, the initial state of charge of on board energy source C_{init} with maximum capacity $C_{\max}$, $(C_{\max}, C_{\text{init}} \in R^+)$, the energy optimal routing problem is to find a path P^j in* **G** *from* **s** *to* **t** *with minimal cost $C(P^j_{s\to t})$ (or equivalently $C(P^j_{v_0^j\to v_N^j})$) under the road segment energy cost given by (12).*

The solution of this optimal routing problem, which depends on the traffic state as well as on the route profile, does not consider the driving time to destination. Generally the time to destination is an important parameter for all the drivers ant it cannot be neglected. We propose to consider it as a constraint to the optimization problem which value depends on driver particular situation. In this manner we calculate the energy optimal route to destination respecting a given time to destination constraints. Solution feasibility of this problem depends on the time constraint as well as on state of charge of batteries and super capacitors. There are also situations in which the driver's primary criterion is the travelling time to destination. To consider it in a consistent way we can formulate the corresponding optimal routing problem consisting in minimization of time to destination with respect to on board energy constraints. So we can define a generic optimization problem including these two aforementioned optimization problem as particular cases:

Constrained optimal path (COP) problem formulation: Consider the graph $G = (V,E,w)$. Each edge $(v_i,v_j) \in E$ is specified by a link weight vector with m additive edge weights $w_l(v_i,v_j) \ge 0, \forall l, 1 \le l \le m$. Given m constraints $L_l (1 \le l \le m)$ the problem is to find a path $(P^j_{s\to t})$ from a source node **s** to a destination node **t** such that the path cost $c(P^j_{s\to t})$ is minimized under the following constraint:

$$w_l(P^j_{s\to t}) = \sum_{(v_i,v_k)\in P^j_{s\to t}} w_l(v_i,v_k) \le L_l, (1 \le l \le m).$$

Remark 1: The weight vector $w(v_i,v_j)$ in the *COP* problem formulation represents the road segment (v_i,v_j) travelling time or rescaled energy one. Whereas the cost represents the route to destination energy or time respectively. Other types of constraints may be introduced giving rise to the multi-constrained optimal path (*MCOP*) problem treated in [18]. Because of lack of space and in order to remain consistent with our objective we will focus on the energy optimal route calculation with time constraints.

We can deduce from the COP problem formulation that road segment energy cost and time constraint function as well as the path time constraints function are given by:

$$\begin{aligned} c(v_i,v_j) &= c_e^m(v_i,v_j) \\ w(v_i,v_j) &= c_t(v_i,v_j) \\ L &= \beta c_t(P^{SP}_{s\to t}). \end{aligned} \tag{13}$$

In (13) $c_t(v_i,v_j)$ is the time cost travelling of road segment and $c_t(P^{SP}_{s\to t})$ is the shortest path length from source node $\mathbf{s}$ to the target node $\mathbf{t}$ for the time cost function. It can be equally understood as the shortest possible time a driver can get from $\mathbf{s}$ to $\mathbf{t}$ which will be called in the sequel as time *lower bound*. The maximum time limit is denoted as L and it is also called time *upper bound* or time constraint. The β is a parameter which allow us to obtain L by multiplying it with the time *lower bound*. Simply said the β tell us how much more time the driver is willing to spend relatively to the *time minimal path*. The values of $\beta \in (1,\infty)$ define the objective of the routing problem. If the value of $\beta = 1$ we have a time optimal routing problem. On the other hand, if $\beta = \infty$ we have an energy optimal routing problem.

IV. Energy Optimal Time Constrained Route Calculation

We propose a new algorithm that finds a path whose energy cost is optimal and also the time cost is at most β times the cost of the time optimal path. The algorithm is based on a combination of ideas proposed in [17], [18], [10]. The core algorithm is based on the [18] and [17] algorithms, which are a modification of *Dijkstras* algorithm that finds *k shortest paths*, where not only the optimal path but also many candidate paths are tentatively stored. Additionally, we combine the A^* search speed-up technique for which we calculate the lower bounds with a classic *Dijkstras* algorithm going backward. During the relaxation step we employ pruning techniques called *path dominance* and *look-ahead*, both effectively reduce the number of paths that would otherwise need to be explored. These techniques are explained in the following subsections.

A. Time Lower Bounds Computation for the Look-Ahead Concept

The look-ahead concept can be viewed as an additional mechanism to reduce the search space of possible paths. The idea, is to limit the set of possible paths by using information of the remaining subpath towards the destination. The look-ahead concept proposes to compute the shortest path tree rooted at the destination to each node in the graph. With respect to *COP* problem formulation, the key insight of look-ahead is to provide each node with a lower bound of exact time cost. Considering this description of the look-ahead concept from [18], we have to realize that we cannot preprocess the lower bound for all nodes in the graph because simply the graph is too large and it would mean that we have to compute *Dijkstras* distance for the whole graph. Such preprocessing would take more time that the main part of the algorithm. To overcome this problem we have, firstly, used a stopping condition in the *Dijkstras* algorithm which ensured that we do the computation only for those nodes that are reachable within the given time constraint. We start from the destination node $\mathbf{t}$ towards the origin node $\mathbf{s}$. When the origin is settled we store the calculated cost, which in this case represents the minimum time that is needed to travel from $\mathbf{s}$ to $\mathbf{t}$, or in other words, it is the cost $c_t(P^{SP}_{s\to t})$ for the shortest path of the time cost function c_t. However, at this moment we do not stop the search as we normally would, but we keep the computation running until the summed cost $c_t(P_{v\to t})$ for some nodes $v \in V$ is greater than $\beta c_t(P^{SP}_{s\to t})$ or

Algorithm 1. Backward Heuristic exploration by Dijkstra's algorithm.

Input: Directed weighted graph $G = (V, E)$, source vertex s, destination vertex t, cost function c
Output: Lower bounds set h representing estimated accumulated cost to destination t from all vertices $v \in V$ that have cost $c(P_{v\to t}) \le c(P_{s\to t})$

```
procedure DIJKSTRA (G, s, t, c)
    h(t) ← 0;
    Q ← {t};
    while Q ≠ ∅ do
        u ← choose the node with minimal h(u) from Q;
        Q ← Q\u;
        if u = s then
            return h;
        end if
        for all ingoing edges (v, u) of node u do
            h′ ← h(u) + c(v, u);
            if v ∉ Q then
                h(v) ← h′;
                Q ← Q ∪ v;
            else if h(v) > h′ then
                h(v) ← h′;
            end if
        end for
    end while
end procedure
```

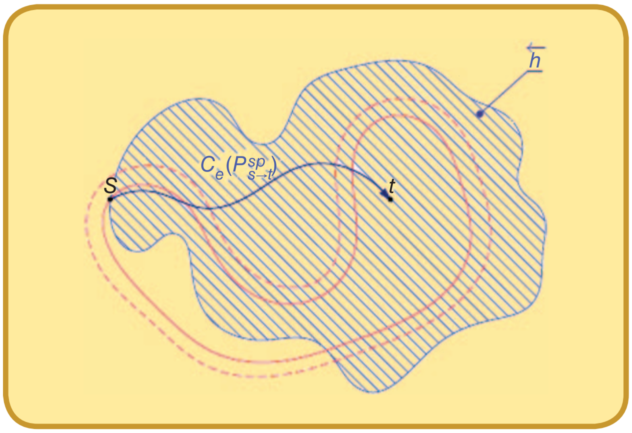

FIG 2 Nodes within the hatched area have their head paths cost computed in the $\overleftarrow{h}$ set. We can see that the search space and the optimal path is different from the time search (see figure 1). This can be explained by the fact that the energy function prefers slower and shorter paths.

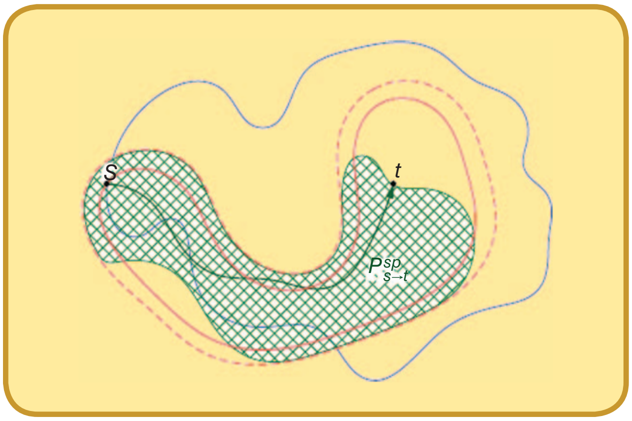

FIG 3 Forward constrained search space.

there are no other unsettled nodes. When the search stops we have a lower bound set that contain an estimated minimal time needed to reach a destination from that node. We denote this set as $\overleftarrow{w}(v)$, which contains the distance for every node v that meets the condition (14) [18], [17].

$$c_t(P_{v\to t}) \leq c_t(P^{SP}_{s\to t}). \tag{14}$$

The idea behind this is that we will compute the distance only for the set of nodes that can eventually be part of the solution. In addition, we are certain that if a node is not included in this solution set then we cannot reach the target node **t** from it within the given time constraint. To calculate the solution set of nodes we use *backward Dijkstras* algorithm based on a priority queue Q that is initialized with the destination node **t**. Each node **u** has associated a tentative distance $h(u)$ from t, which is the only component that decides the priority within the queue. The algorithm works in the same fashion as the classic *Dijkstras algorithm*, but the predecessor list

Algorithm 2. Energy optimal path search with time constraint.

Input: Directed weighted graph $G = (V, E)$, source vertex s, destination vertex t, constraint multiplier β, time cost function c_t, energy cost function c_e

Output: A shortest path between s to t obeying the time cost constraint

procedure FindPath(G, s, t, β, c_t, c_e)
  $\overleftarrow{w} \leftarrow$ Dijkstra(G, s, t, c_t); *noting that* $\overleftarrow{w}(a) = \overleftarrow{w}(P_{a\to t})$
  $\overleftarrow{h} \leftarrow$ Dijkstra(G, s, t, c_e); *noting that* $\overleftarrow{h}(a) = \overleftarrow{h}(P_{a\to t})$
  $\vec{w}(P_{s\to s}) \leftarrow 0$;
  $\vec{h}(P_{s\to s}) \leftarrow 0$;
  $S \leftarrow \emptyset$;
  $Q \leftarrow \{P_{s\to s}\}$;
  while $Q \neq \emptyset$ **do**
    $P_{s\to u} \leftarrow$ choose the path with minimal $\vec{h}(P_{s\to u}) + \overleftarrow{h}(u)$ from Q;
    $Q \leftarrow Q \setminus P_{s\to u}$;
    if $u = t$ **then**
      return $P_{s\to u}$;
    end if
    for all outgoing edges (u, v) of node u **do**
      if $v \notin \overleftarrow{w}$ **then**
        $\overleftarrow{w}(v) \leftarrow \overleftarrow{w}(s)$;
      end if
      if $\vec{w}(P_{s\to u}) + c_t(u, v) + \overleftarrow{w}(v) > \beta \cdot \overleftarrow{w}(s)$ **then**
        continue; *prune this path as it is not feasible*
      end if
      $P_{s\to v} \leftarrow$ extend $P_{s\to v}$ with edge (u, v);
      $\vec{w}(P_{s\to v}) \leftarrow \vec{w}(P_{s\to u}) + c_t(u, v)$
      $\vec{h}(P_{s\to v}) \leftarrow \vec{h}(P_{s\to u}) + c_e(u, v)$;
      if IsDominated($S, \vec{h}, \vec{w}, P_{s\to v}$) **then**
        continue; *prune this path as it is dominated*
      end if
      Path $P_{s\to v}$ *is considered feasible so it is added to the queue.*
      if $v \notin \overleftarrow{h}$ **then**
        $\overleftarrow{h}(v) \leftarrow \overleftarrow{h}(s)$;
      end if
      $Q \leftarrow Q \cup P_{s\to v}$;
    end for
  end while
end procedure

is not stored, because we are only interested in obtaining the distances from all vertices $v \in V$ that have cost $c_t(P^{SP}_{v\to t}) \leq c_t(P^{SP}_{s\to t})$ to destination node t. This procedure is given in *Algorithm* 1.

B. Energy Heuristic Computation

In order to speedup the energy minimal path calculation we use A^* algorithm for which the lower bounds of path energy to destination are calculated using backward *Dijkstras* algorithm. The total or projected energy cost is given by $\vec{h}(P_{s\to v}) + \overleftarrow{h}(v)$ representing the energy spent from source node **s** to the current node **v** and an evaluation of the energy cost to destination node **t** respectively. Since we are using the *Dijkstras* distances as the heuristic, we can use identical computation as the one used for the time lower bounds in the previous section. We are using the same procedure given in Algorithm 1. The difference is, that instead of the time cost function c_t,

Algorithm 3. Path dominance test.

Input: Directed weighted graph $G = (V, E)$, set $\vec{w}$ representing time spent on path P from source vertex s to intermediate vertex v, set of non-dominated paths S

Output: Returns 1 if the path $P^{*}_{s\to v}$ cannot be a subpath of a shortest path, otherwise 0

```
procedure IsDominated(S, h⃗, w⃗, P*_{s→v})
    for all paths P^i_{s→v} from set S do
        if h⃗(P^i_{s→v}) ≤ h⃗(P*_{s→v}) and w⃗(P^i_{s→v}) ≤ w⃗(P*_{s→v}) then
            return 1;
        end if
    end for
    S ← S ∪ P*_{s→v};
    return 0;
end procedure
```

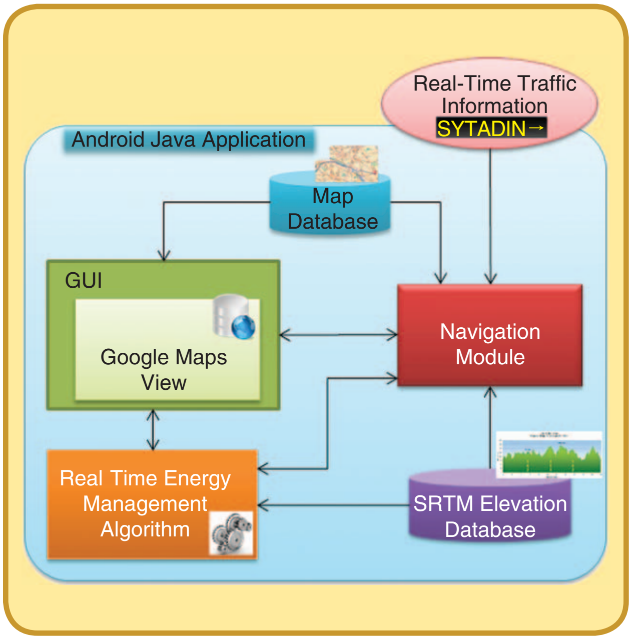

FIG 4 Architecture of the Energy Optimal Real Time Navigation System.

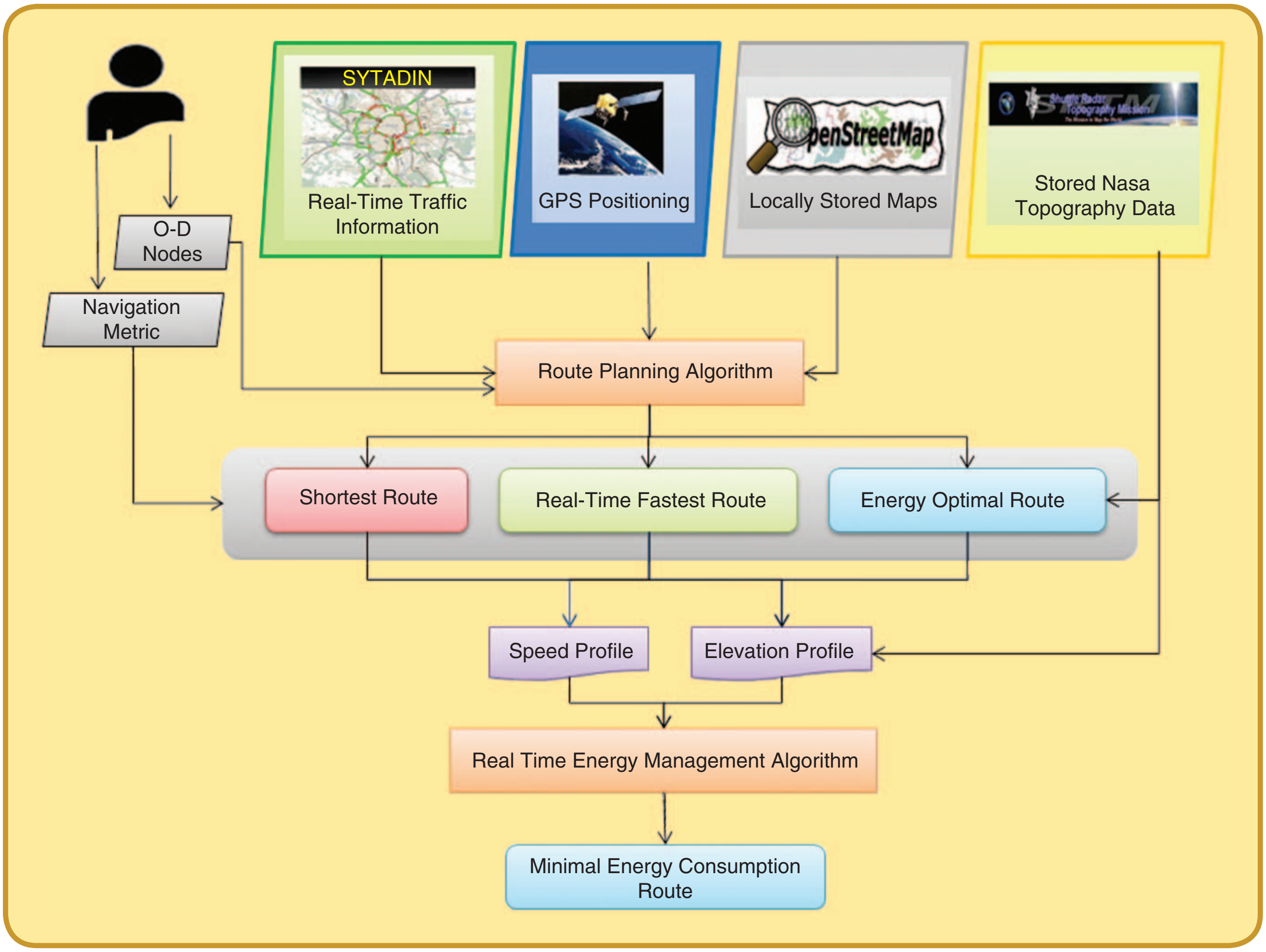

FIG 5 Energy Optimal Real Time Navigation System application flow-chart.

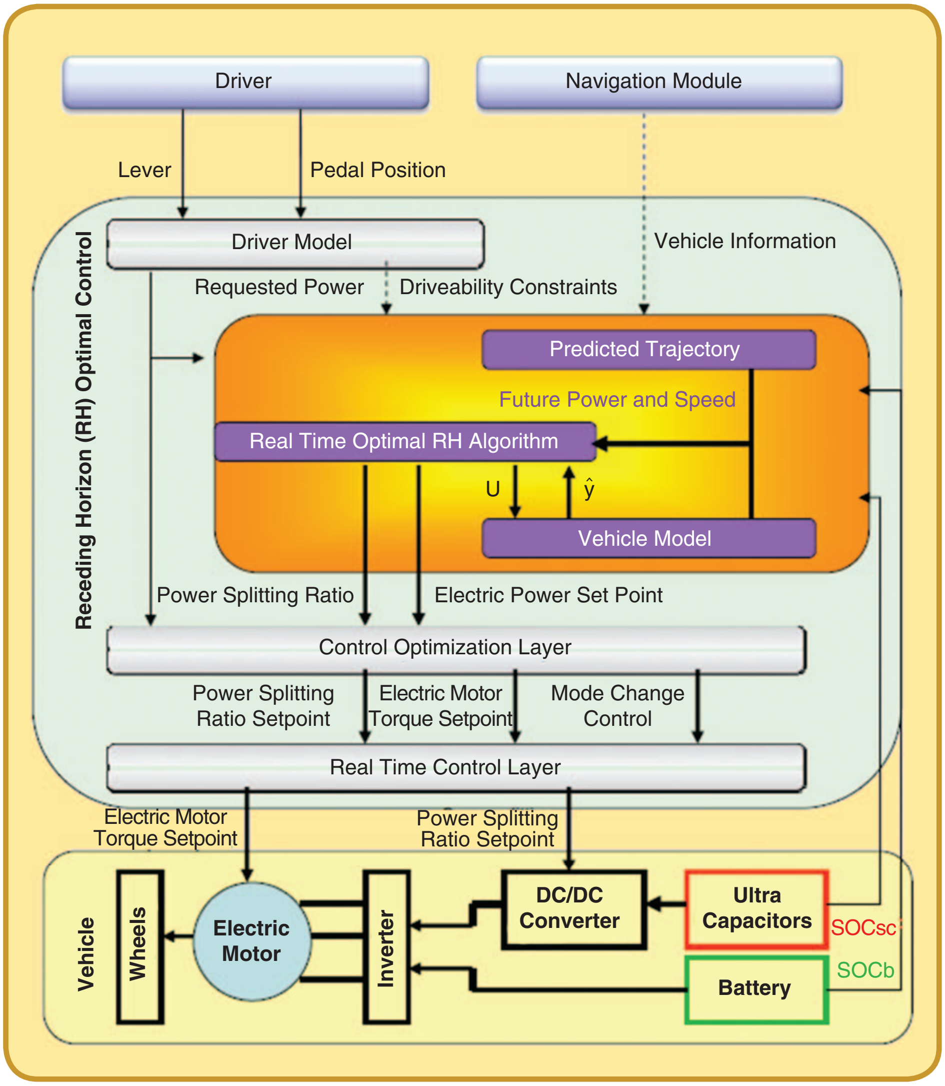

FIG 6 The R^T *EMA* data and functional flow diagram.

we are using the energy cost function c_e^m. Although, we cannot use the energy cost function as it is defined in (12), because this search goes backward from the destination node **t** and we do not know what state of charge of the energy source will be at the destination. We only know the state of charge at the origin node and we can calculate the state of charge for other nodes only after the energy optimal path is found.

Therefore, we need to remove the dependence on the state of charge. We know that the state of charge is used to penalize the states of *overcharging* or *discharging* limit of on board source energy. Additionally, we know that this two extreme states can only increase the cost, so we decide to ignore these two extreme states and the possible increase of cost. Moreover, we can prove that such a modification is consistent with an admissible heuristic due to the fact we do not overestimate the projected path cost. In figure 2 an example of the admissible search space corresponding to energy metric is given.

The third step of the algorithm is the forward search where the time constrained optimal path is found. We recall that a given path is admissible if sum of time spent on the tail path $\overrightarrow{w}(P_{s\to v})$ and the estimated cost of the head path $\overleftarrow{w}(v)$ (the path from the node **v** to the target node **t**) is lower than the time constraint given by (15). Furthermore, the candidate paths are ordered properly such that the path with lowest projected cost $\overrightarrow{h}(P_{s\to v}) + \overleftarrow{h}(v)$ is selected and expanded first which allows to terminate our expansion procedure once we have found the constrained shortest path satisfying the time constraint (15) or there are no candidate paths left.

$$\overrightarrow{w}(P_{s\to v}) + \overleftarrow{w}(v) \leq L. \quad (15)$$

On figure 3 we can see how the resulting path might look like regarding the pre-processing steps. The search space is hatched in green and we can notice that it never crosses the red dashed line, which represents the time constraint boundary that was settled by the time lower bounds pre-processing step. We can claim that if any node is located outside of this boundary can not be part of the feasible solution because it do not satisfy the time constraint, and so if any path attempts to expand over this boundary it is pruned. In reality the pruning happens even in larger distance from this border due to the look-ahead concept. This procedure is given in *Algorithm* 2.

C. Path Dominance

Path dominance is a speed-up technique generally used to reduce the calculation complexity. In order to explain this technique we give the following example. Suppose we have reached some intermediate node v by path P_A and the time needed is 15 minutes. However, if there is some other path P_B, which also leads to the node v using only 12 *minutes* and the same amount of energy as path P_A, then we do not

Table 1. Electrical vehicle parameters.

Parameters	Value
M (nominal mass)	1680.0 kg
M (total mass)	2095 kg
r_t (Tire radius)	290 mm
C_d (reynolds coefficient)	0.32 kg/m^3
A_f (windward area)	2.31 m^2
μ_r (rolling resistance coefficient)	0.0088
ρ (air density)	1.25 kg/m^3
δ_{eqm} (equivalent moment inertia)	0.195 kg m^2
Power of electric motor (nominal)	45 kW
SC bank	2 $\times$ 66 series connection SCs
Battery pack	14 series connection batteries

FIG 7 Shortest (red color), Fastest (blue color) and Energy optimal route (green color). The origin point corresponds to the red tag and the destination point corresponds to the violet one. The coefficient of energy recuperation α used is equal to 0:85.

need to store the path P_A, because we already know that it will lead to same or lower quality paths with respect to path PB. By quality of path we understand the amount of used resources needed to follow this path. Such path is called *dominated path* and it is pruned. The test of path dominance is given by Algorithm 3.

V. Software Architecture Solution

Figure 4 and figure 5 show the main component and application structure of the *EORTNS* software architecture. The core of the system is composed of four main modules: *Navigation* module, *Energy Optimization* module, *Graphic User Interface* module and *Real Time Energy Management Algorithm ($R^T EMA$)* [12] module. The *Graphic User Interface* module handles the interaction with user, draws the optimal calculated route on the map and displays calculation progress and results of the optimization.

The *Navigation* module operates with the map database from which the traffic area directed graph is retrieved. The static nodes and route segment information are enhanced with the traffic real time information like velocities and time duration for each road segment. Moreover, information such as speed and 3*D* elevation profile are calculated for each discretization points of each calculated route to destination segments. This information is necessary to obtain the route to destination power profile in order to operate real time power splitting ratio between battery and *SC* from the $R^T EMA$ module as given in figure 5.

Table 2. Shortest, fastest and energy optimal route parameters.

Metric	Color	Path Length	Duration	Arriving Battery SOC_b
Shortest	Red	13.8 km	21 min	94.9 %
Fastest	Blue m	16.6 km m	15 min	92.2 %
Energy optimal	Green	14.3 km	23 min	95.4 %

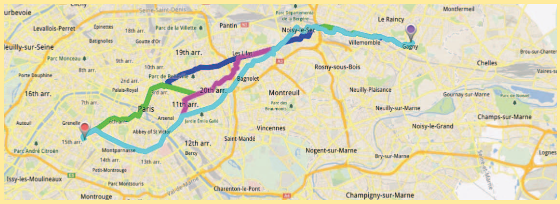

FIG 8 Energy optimal routes for different values of recuperation coefficient α. The chosen values of α as well as the color code are those given in figure 10. We note that these routes have different parts in common. The origin point corresponds to red tag and the destination point corresponds to violet one.

A. Real Time Energy Management Algorithm $R^T EMA$

The $R^T EMA$ software module, given in figure 6, makes part of *EORTNS* and is a software module proposed in the previous release [12]. As we can see, from the *navigation module* we obtain real time route to destination which are composed of road segment velocities and their 3D geo-coordinates. These data are used to calculate the power profile of the *HEV* vehicle for each route based on their longitudinal model. From the power demanded profile we calculate the energy losses in batteries and in Super-Capacitors as well as in motor based on their proper characteristics. The $R^T EMA$ software module addressed the minimization of these losses under the constraints imposed by the dynamic of the vehicle, the model of battery and Super-Capacitors by combining the so called receding horizon philosophy [20] and *Dynamic Programming algorithm* [21], [22]. The results obtained are the power splitting ration as well as the gear ratio of gearbox which are furnished at the lower real time levels responsible to distribute the power needed between batteries and Super-Capacitors. We can find more details on $R^T EMA$ software module in [12].

FIG 9 Found paths for different values of parameter β.

B. 3D Maps and Routing Data-Base

The geographic data is exported from the *OpenStreetMap* (*OSM*) server, which provides free data distributed under the *Creative Commons CC BY-SA 2.0* license.

1) Routing Map Export

The *OSM* map is exported into a binary file with a simple structure which consist of several tables with records. Each record has fixed byte length so we can easily seek in the file with only knowing the **id** to find the right record. During the export, we store only the relevant information for road routing and the rest is ignored and we store the map in a format that is suitable to describe a directed graph. Three tables are created: nodes, edges and geometry. Table with nodes contains only the junctions nodes or nodes where some characteristic of the road change (e.g. road class, speed,...) and new edge should follow. In the edge table we store the information that is needed to calculate the cost to travel this path. Finally, in the geometry table we store intermediary nodes that describe the shape of the road. They are not necessary for the routing but are used for the energy optimization algorithm and for drawing of the road in the GUI.

The resulting routing database for the *Ile-de-France* region has 36 MB.

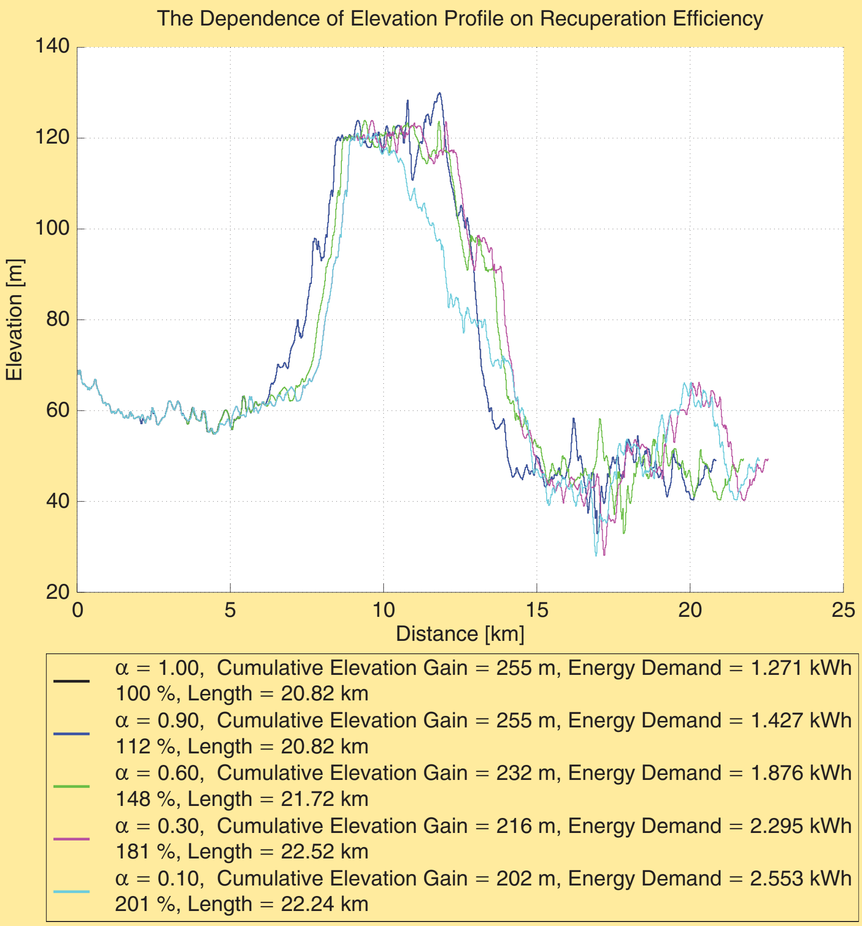

FIG 10 The dependence of elevation profile on energy recuperation coefficient α. The cumulative elevation gain or the route elevation profile variation as well as the route energy reduction increase with the value of α. For $\alpha = 1$ and $\alpha = 0.9$ the routes to destination calculated are identical.

2) Exported 3D Map

In order to add the third dimension to the exported *OSM* map we have two possibilities: retrieve the altitudes of a calculated route with an on-line elevations service request or with a device stored elevations file. The first was the initial solution to obtain the altitude profile necessary for the power profile used by $R^T EMA$ module. The second is a more complicated solution since it consists of retrieving *NASA's Shuttle Radar Topography Mission* (*SRTM*) generated files, treating them and generating a permanent data base for the application. The advantages of the second option are Internet independence and faster elevations retrieval for given coordinates. A single request contains a list of all geocoordinate elements. For *SRTM*, the binary files provided by *NASA* are divided by tiles of one degree per one degree of latitude and longitude.

> The value of the energy recuperation coefficient plays an important role on the energy efficiency as well as on the altitude profile of the calculated route.

In our implementation we currently use a five to five degree tile that is created by combining the individual one degree *SRTM* files. Most of the time we want to know an elevation of points that are not exactly positioned at one of the known nodes of the elevation grid so in order to calculate such elevation we use a bi-cubic interpolation.

C. Real Time Traffic Information

As a real-time traffic information source, we use *SYTADIN* web service which provides traffic information for the *Paris* and *Ile-de-France* region. First, we have attributed the *SYTADIN's* traffic *IDs* to our *OSM* nodes in the map database by parsing *SYTADIN's MapInfo* traffic net geometry file. The *SYTADIN* web service provides an updated *XML* file with traffic information of each road segment identified by it's *ID* every few minutes (3 to 5 minutes). This way, we can obtain instant segment speed or traffic state (fluid, traffic-jam, closed) for every node in our database.

VI. Application Example

A demonstration application was developed including all the aforementioned modules. For the computation device we use the *Samsung Galaxy Tab 10.1* tablet, which embeds *NVIDIA Tegra 2* dual-core *1GHz* processor, *1GB RAM*, *16 GB* of internal storage and also offers a 10.1 inch touchscreen, Wi-Fi, *GPS* and *3G* connection.

A. Implementation, Simulation and Analysis

The prototype HEV vehicle to carry out the simulation is the *DaimlerChrysler* [3] car. The power sources are composed of 2×66 series connection *SCs* [13] and a branch of 14 series connection *90Ah* [13]. The parameters of the prototype vehicle are listed in Table 1.

In figure 7 the routes calculated for the three metrics used are given. The route characteristics in term of, length, time duration and the state of charge (SOC_b) of battery at the target point are given in Table 2. It is interesting to calculate the energy gain ratio obtained from energy metric algorithm with respect to the distance metric one. We can easily observe that for this application test, the energy gain ratio is about 10% which is far from negligible.

Naturally the value of energy recuperation coefficient, α, plays an important role on the energy efficiency as well as on the altitude profile of the calculated route. Its value depends on several factors and more specifically on the characteristics of the power train architecture as well as on the road surface condition [14]. In Figure 8, for different values of α, we give the optimal routes calculated using energy metric. In figure 10 we give the different values of α used and the associated color codes corresponding to different optimal routes. In the same figure, we give also the cumulative elevation gain, energy demand and the length of the calculated route. The cumulative elevation gain is an indicator of the route elevation variation. We may observe from figure 10 that the route elevation variation increases with the increase of α value.

In Figure 9 we can see four calculated paths for different values of parameter β. The route characteristics in term of length, duration and the state of charge at the destination point are given in table 2. For this test we take $\alpha = 0.8$ for all paths. The β parameter is set

Table 3. The dependence of elevation profile on recuperation efficiency.

α	Color	Path Length	Duration	Arriving SOC_b
1.0	Black	20.8 km	52 min	93.8 %
0.9	Blue	20.8 km	52 min	93.5 %
0.6	Green	21.7 km	60 min	92.8 %
0.3	Magenta	22.5 km	46 min	91.9 %
0.1	Cyan	22.8 km	41 min	91.3 %

Table 4. Route parameters for different values of parameter β.

β	Color	Path Length	Duration	Arriving SOC_b
1.00	Red	22.2 km	18 min	88.7 %
1.10	Green	20.1 km	19 min	91.1 %
1.15	Cyan	20.2 km	20 min	91.4 %
1.30	Magenta	16.3 km	23 min	93.2 %
∞	Blue	15.0 km	27 min	94.9 %

different for each of those paths, so we can observe the influence it has on the resulting path. For $\beta = 1$ the problem is the same as we would look for the fastest path, because it is the only that can meet the constraint. On the other hand, if $\beta = \infty$ then the path is the energy optimal path regardless the constraint. Three other paths with the time constraint increased by 10%, 15% and 30% are given. We can observe that with increasing β the duration of the path and the arriving state of charge increases. It means that the time constraints relaxation induces slower *energy optimal path* which is in concordance to general intuition.

VII. Conclusions and Perspectives

In this paper an *energy optimal real time navigation system* (*EORTNS*) is proposed focused mainly on energy optimal route calculation. It constitute a new release of Optimal Real Time Navigation System (*ORTNS*) [12] and is based on the real time traffic information system provided by *SYTADIN*. The *EORTNS* offers *HEV* real time energy optimal navigation which consists in optimal and parametrized route calculation as well as optimal power splitting ratio between battery and *SC*. It is fully independent and capable of retrieving real time traffic network information from *SYTADIN* and operate real time route to destination updates.

From the simulation test presented in this paper we can clearly see that the energy metric helps in on board energy optimization and the results obtained are better than those obtained based on distance metric. Another advantage of *EORTNS* is related to the increases of battery life operated through the reduction of battery charging cycles [12]. We observe also the importance of the energy recuperation coefficient α in route calculation as well as in on-board energy optimization. This fact gives us a clear direction of research in order to increase the energy efficiency of *HEV* as well as of other types of hybrid electrical vehicles.

From the Fig. 9 we can see clearly that relaxing the time to destination constraints (increase the value of β) operates a modification of path to destination and reduces the energy consumption. The extra time we have to reach the destination can be considered as a supplementary resource at our disposal helping to further optimize the energy consumption. We can go further with this reasoning and say that it may help also to reduce the congestion state in some limited zone of traffic, naturally if the drivers are aware of and they accept to spent more time in traffic. This pushes us toward the approaches integrating an organized collaboration between drivers as one of the key factors in minimizing time and energy in traffic network by reducing *the price of anarchy* [19] we pay applying individual/selfish routing strategies/algorithms.

The future development of *EORTNS* will be focused on the integration of other traffic network information sources and collaborative real time route state estimation in order to enhance the information provided by *SYTADIN*, increase the accuracy of route velocity profile as a key factor in the design of efficient routing strategies/algorithm based on drivers collaboration.

Acknowledgment

This research work is partially supported by AKKA Technologies (http://www.akka.eu/), our industrial partner in this project, that we warmly thank for the support and the confidence which we have granted.

About the Authors

Tomas Jurik received the Bc. and Ing. degrees in the field of cybernetics, control and measurements from the Faculty of Electrical Engineering and Communication at Brno University of Technology, Brno, Czech Republic, in 2010 and 2013, respectively. He joined the project team in September of 2012 and worked on the *Energy Optimal Navigation System* during his Erasmus exchange at the UPE, ESIEE Paris over the course of the academic year 2012/2013.

Arben Cela received the B.S. and Ing. degrees from the Polytechnic University of Tirana, Albania in 1984. From 1984 to 1986 he was with the Department of Power Generation of Faculty of Engineering of Polytechnic University of Tirana, Albania. He received the Ph.D. degree and and French Habilitation (HDR) from Université Paris-Sud respectively in January 1993 and in June 2011. He is currently with Laboratoire des Signaux et Systèmes (L2S, UMR CNRS 8506), CNRS-Supélec, Gif-Sur-Yvette 91192, and with Computer Science and Telecommunication Department of ESIEE Paris, UPE, Noisy-Le-Grand 93162, France. He is author/coauthor of one book and of more than 85 conference and journal papers in the area of optimization, control of distributed real time systems and delayed systems.

Redha Hamouche was born in Algiers in 1974. He received the Ph.D. degrees, in Real-time Embedded Systems, from the University of Evry, France. He joined in 2003 Universit´e Paris Est, ESIEE Paris, where he is currently associate professor at ISYS Department. His research interests include modeling and design of Real Time Embedded Systems with special focus on Navigation Systems for Hybrid Electrical Vehicle.

René Natowicz received the Ph.D. degree in Computer Sciences from the University of Paris-Sud, Orsay, France, in 1987. He is associate professor at ESIEE Paris, Computer Science and Telecommunication Department, UPE, Noisy-Le-Grand 93162, France. His research interests include combinatorial optimization, machine learning, bioinformatics. He has more than 50 publications in journals and conferences in these areas.

Abdellatif Reama was born in Morocco in 1958. He received the Ph.D. degree in electrical engineering from National Polytechnic Institute of Toulouse, France, in 1987. He is currently an Associate- Professor in Electrical Engineering at Universit´e Paris Est, ESIEE Paris. His research interests include several aspects of the Electrical Engineering with special focus on power electronics and control design.

Silviu-Iulian Niculescu received the B.S. degree from the Polytechnical Institute of Bucharest, Romania, the M.Sc., and Ph.D. degrees from the Institut National Polytechnique de Grenoble, France, and the French Habilitation (HDR) from Université de Technologie de Compiègne, all in Automatic Control, in 1992, 1993, 1996, and 2003, respectively. He is currently Research Director at the CNRS(French National Center for Scientific Research) and the head of L2S (Laboratory of Signals and Systems), a joint research unit of CNRS with SUPELEC and Université Paris-Sud located at Gif-sur-Yvette. He is author/coauthor of 5 books and of more than 375 scientific papers. His research interests include delay systems, robust control, operator theory, and numerical methods in optimization, and their applications to the design of engineering systems.

Jérôme Julien is graduated from the *Université de Technologie de Compiégne* in mechanical engineering. Beginning his career in industrial design, he joined *Akka Technologies* in 2008 in the research and development center of the group, *Akka Research*. Since, his leading research project is the mobility field with particular focus on automotive applications.

References

[1] D. B. Work and A. M. Bayen, "Impacts of the mobile internet on transportation cyberphysical systems: Traffic monitoring using smartphones," in *Proc. Nat. Workshop Research High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation and Rail*, Washington, DC, Nov. 18–20, 2008.

[2] A zero infrastructure traffic information system. [Online]. Available: http://trans.civil.northwestern.edu/thanasis/ research/projects/zi.html

[3] L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modelling and Optimization*. New York: Springer, 2007.

[4] P. Sanders and D. Schultes, "Highway hierarchies hasten exact shortest path queries," in *Proc. 13th Annu. European Conf. Algorithms*, 2005, pp. 568–579.

[5] D. G. Thomas, *Fundamentals of Vehicle Dynamics*. Warrendale, PA: Society of Automotive Eng., 1992.

[6] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction hierarchies: Faster and simpler hierarchical routing in road network," in *Proc. 7th Int. Conf. Experimental Algorithms*, 2008, pp. 319–333.

[7] D. B. Johnson, "Efficient algorithms for shortest paths in spare networks," *J. ACM*, vol. 24, no. 1, pp. 1–13, 1977.

[8] J. Eisner, S. Funke, and S. Storand, "Optimal poute planning for electric vehicles in large networks," in *Proc. 25th AAAI Conf. Artificial Intelligence*, San Francisco, CA, Aug. 2011.

[9] S. Storandt, "Algorithms for vehicle navigation," Ph.D. dissertation, Faculty of Comput. Sci., Elect. Eng. Inform. Technol., Univ. Stuttgart, Stuttgart, Germany, Dec. 2012.

[10] M. Sachenbacher, M. Leucker, A. Artmeier, and J. Haselmayr, "Efficient energy-optimal routing for electrical vehicles," in *Proc. 25th AAAI Conf. Artificial Intelligence*, San Francisco, CA, Aug. 2011.

[11] Sytadin: A real time traffic information of paris area. [Online]. Available: http://www.SYTADIN.fr/

[12] A. Hrazdira, A. Cela, R. Hamouche, A. Reama, B. Rezende, S.-I. Niculescu, and C. Villedieu, "Optimal real-time navigation system: Application to a hybrid electrical vehicle," in *Proc. 15th IEEE Int. Conf. Intelligent Transportation Systems*, Anchorage, AK, Sept. 16–19, 2012, pp. 409–414.

[13] Société Batscap. [Online]. Available: http://www.batscap.com/

[14] D. Yin and Y. Hori, "A novel traction control for electric vehicle without chassis velocity," in *InTechOpen*. 2010.

[15] J. Wang, J. Jiang, and T. Mu, "Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1331–1345, Sept. 2013.

[16] T. M. Sweda and D. Klabjan, "Finding minimum-cost paths for electric vehicles," in *Proc. IEEE Electric Vehicle Conf.*, 2012, pp. 1–4.

[17] G. Liu and K. G. Ramakrishnan, "A* Prune: An algorithm for finding K shortest paths subject to multiple constraints," in *Proc. 20th Annu. Joint Conf. IEEE Computer Communications Societies*, 2001, pp. 743–749.

[18] P. V. Mieghem and F. Kuipers, "Concepts of exact QoS routing algorithms," *IEEE/ACM Trans. Networking*, vol. 12, no. 5, pp. 851–864, Oct. 2004.

[19] T. Roughtgarden and E. Tardos, "How bad is selfish routing?" *J. ACM*, vol. 49, no. 2, pp. 236–259, Mar. 2002.

[20] E. Camacho and C. Bordons, *Model Predictive Control*. London, UK: Springer-Verlag, 2004.

[21] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. I, Belmont, MA: Athena Scientific, 1995.

[22] E. Denardo, *Dynamic Programming: Models and Applications*. Englewood Clifs, NJ: Prentice-Hall, 1982.

ITS