A better alternative to dynamic programming for offline energy optimization in hybrid-electric vehicles

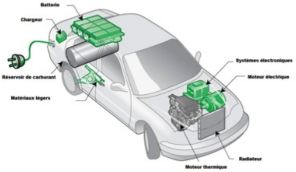
Sandra Ulrich NGUEVEU
Mouloud Guemri, Stéphane Caux, Frédéric Messine

LAAS-CNRS / LAPLACE / IRIT
ngueveu@laas.fr, {guemri, caux}@laplace.univ-tlse.fr

24/06/2013

Plan

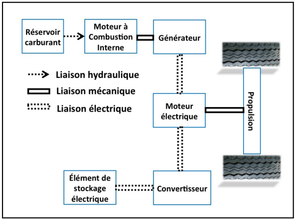
Hybrid-electric vehicles

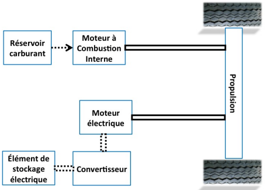


Electric propulsion motor powered by :

- onboard generator :
  - internal combustion engine or
  - hydrogen fuel cell (FC)
- reversible source :
  - battery or
  - supercapacitator (SE)

Architectures (hybrid-series, hybrid-parallel, ...)

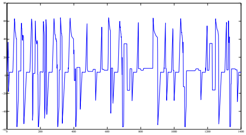




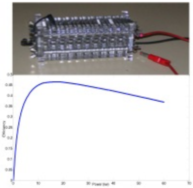
Problem description

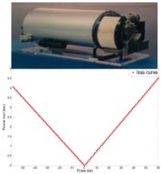
Given

- the power request of a driver on a predefined road section



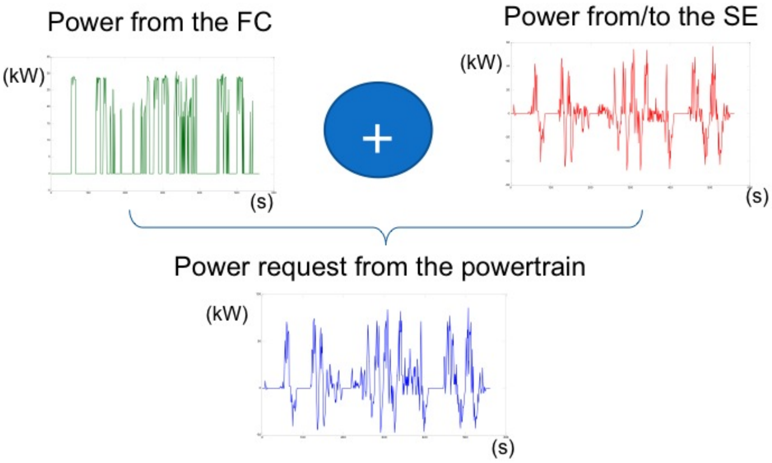
- the characteristics of the energy sources : power limitations (kW), efficiency (%), storage capacity (kWs)





Find at each instant the optimal power split between the energy sources to satisfy to minimize the total fuel consumption.

Example of solution



Mathematical formulation

Input data

- r : power request
- f_c : FC efficiency function
- f_B : SE efficiency function

Decision Variables

- u : power from FC
- $v = r - u$: power from SE
- x : SE state of charge

$$\min_u \int_0^T f_c(u(s))ds \quad (1)$$

$$s.t. \quad \dot{x} = -f_B(x, r-u) \quad (2)$$

$$x(0) = x_0, x(T) = x_f \quad (3)$$

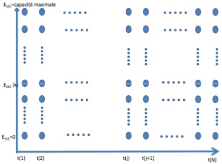
$$0 \leq u \leq u_{\max} \quad (4)$$

$$r - K_{\max} \leq u \leq r - K_{\min} \quad (5)$$

$$x_{\min} \leq x \leq x_{\max} \quad (6)$$

Literature review (2012*)

Dynamic Programming (DP) requires an additional discretization of the energy levels



Several faster approaches have been proposed in the literature ...

- ECMS, Linear Programming, Shooting techniques, Rule-based Programming, Genetic algorithms

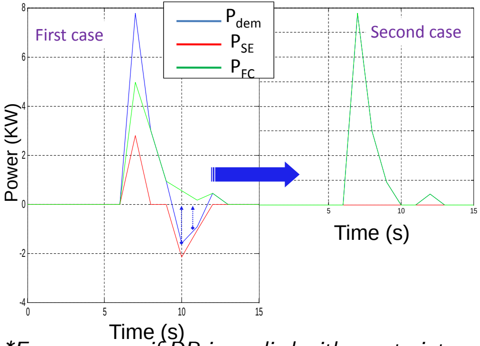
... but none of them found solutions of better quality than DP.

Hence the widespread perception that :
DP provides the optimal solution of such energy management problems.

Weaknesses of Dynamic Programming

Recovery of braking energy

Contrary to the common assumption, imposing that all braking energy has to be recovered can lead to worse solutions*



First case = 57 kWs
Second case = 47 kWs

*Even more so if DP is applied with constraints on the final SOC.

Weaknesses of Dynamic Programming

Equality between the initial and the final state of charge (x(0) = x(N)) is usually imposed :

- to facilitate fair comparisons between different solutions
- to ensure that the vehicle can perform repetitive missions

Final State Of Charge of the SE

In some cases the vehicle had to "burn" excess energy towards the end of the profile to return the energy level to its initial state, which means using the FC at low efficiency levels

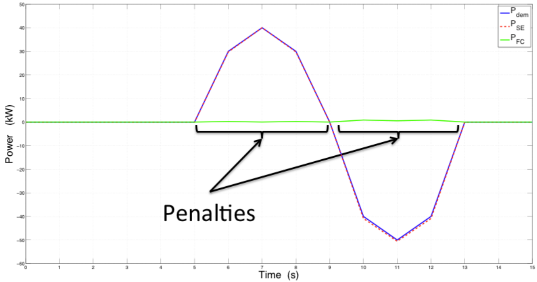
Solution proposed

- impose $x(N) \geq x(N)$ but with no special reward for the additional energy.

Weaknesses of Dynamic Programming

Penalties due to SE discretization

Since the power levels have been discretized, the system tends to respond too strongly to demands that are not exact multiples of the discretization step

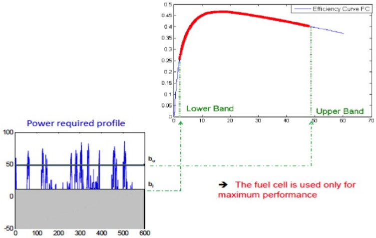


Algo A : Power filtering on the fuel cell

Limits the range of usage of the FC (contrary to "Thermostat")

2 parameters

- lower band B_l : $0 \leq B_l \leq u_{\max}$
- Upper band B_u : $B_l \leq B_u \leq u_{\max}$



Algorithm

- if $r_i \leq B_l$, then set $v_i = r_i$ and $u_i = 0$
- if $B_l \leq r_i \leq B_u$, then set $u_i = r_i$ and $v_i = 0$
- if $r_i \geq B_u$, then set $u_i = B_u$ and $v_i = r_i - B_u$

Algo B : Subgradient optimization - based local search

Reformulation using the natural time-discretization :

(i) $u_i \geq 0$: power generated by FC at time i
(ii) v_i : power generated ($v_i \geq 0$) or stored ($v_i \leq 0$) by SE at time i.
(iii) x_i : state of charge of SE at time i

$$\min \sum_{i=1}^{n} f_c(u_i) \tag{7}$$

$$s.t. \quad u_i + v_i \geq r(i), \quad \forall i \in [1, ..., n] \tag{8}$$

$$\sum_{i=1}^{n} (v_i + \tilde{\rho}(v_i)) \geq 0 \tag{9}$$

$$\sum_{k=1}^{i} (v_k + \tilde{\rho}(v_k)) \leq x_0 - x_{min}, \quad \forall i \in [1, ..., n] \tag{10}$$

$$\sum_{k=1}^{i} (v_k + \tilde{\rho}(v_k)) \geq x_0 - x_{max}, \quad \forall i \in [1, ..., n] \tag{11}$$

Algo B : Subgradient optimization - based local search

Principle

Local search using a Quasi-Newton subgradient optimization method taking into account the non-linear constraints.

Finding the best starting point is key

- starting point = vector of n instants
- random $\rightarrow$ poor results and high computational times

2 variants depending on the starting point

- Algo B^r
- Algo B^A

Lower bounding procedure

Objective

Evaluate the gap between the best solutions from the heuristics and the optimal solution.

Principle

Assuming that $f_c(x_i) = \max_{x_i}\{f_c(x_i)\} = \alpha$ (constant), the objective-function becomes the following linear function :

$$\min f_{\mathrm{LB}}(x_i) = \frac{x_i}{\alpha}. \qquad (12)$$

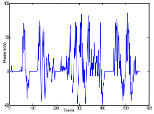
$\Rightarrow$ The resulting objective-function is convex.

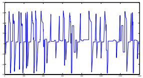
Implementation and Instances

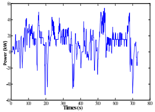
Implementation

- Matlab Version 7.9.0.529 (R2009)
- Intel(R) Pentium(R) 4 CPU 2.80GHz
- RAM : 3 Gb

Instances







- Vehicle : characteristics provided by LAPLACE
- Power request profiles
  1. INRETS (from IFSTARR)
     - electric vehicles in urban environment
     - 561s
  2. ESKISEHIR (from ALSTOM)
     - Turkish tramway
     - 1400s
  3. HIGHWAY (from LAPLACE)
     - vehicle on a highway
     - 750s

Computational Evaluation

INRETS

	DP	Algo. A	Algo. B^r	Algo. B^A
Fuel Cost (kWs)	10131	8869	8750	8776
Computing time	22 h	5 s	23 min	4min
Gap to LB	**17.16%**	2.57%	1.19%	1.49%

Table: Result of our heuristics vs DP

Upper bound (FC only) = 14891 kWs
Lower bound (LB) = 8647 kWs

Computational Evaluation

ESKISEHIR

	DP	Algo. A	Algo. B^r	Algo. B^A
Fuel Cost (kWs)	31826	28365	27601	28151
Computing time	52 h	5 s	10h	3733 s
Gap to LB	**17.81%**	5.00%	2.17%	4.21%

Table: Result of our heuristics vs DP

Upper bound (FC only) = 48043 kWs
Lower bound (LB) = 27014 kWs

Computational Evaluation

HIGHWAY

	DP	Algo. A	Algo. B^r	Algo. B^A
Fuel Cost (kWs)	20099	18773	19016	18670
Computing time	48 h	4 s	1.4 h	986 s
Gap to LB	**9.26%**	2.05%	3.38%	1.50%

Table: Result of our heuristics vs DP

Upper bound (FC only) = 23084 kWs
Lower bound (LB) = 18395 kWs

Conclusion

To summarize

- Our heuristics provides very good solutions
- Classical Dynamic Programming does not always produce optimal solutions of the original problem (it can even produce very poor solutions)
- The lower bound we proposed is of good quality

Recent developments

- Efficient linear reformulation of the problem to obtain very good solutions even faster (Y. Gaoua) : http ://www.laas.fr/files/MOGISA/2013-Conf/Yacine-ICORES2013.pdf