Hybrid Dynamic Optimization for Cruise Speeed Control ⋆

Tiago R. Jorge * **João M. Lemos** ** **Miguel S. Barão** *** **Alberto Bemporad** ****

* *INESC-ID, R. Alves Redol 9, 1000-029 Lisboa, Portugal, (e-mail: tiago.nuno.jorge@ist.utl.pt)*
** *INESC-ID/IST, R. Alves Redol 9, 1000-029 Lisboa, Portugal, (e-mail: jlml@inesc-id.pt)*
*** *INESC-ID/Univ. Évora, R. Alves Redol 9, 1000-029 Lisboa, Portugal, (e-mail: mjsb@ramses.inesc-id.pt)*
**** *Dept. of Mechanical and Structural Engineering, Univ of Trento, Via Mesiano 77, 38100 Trento, Italy, (email: bemporad@ing.unitn.it)*

Abstract: The cruise control problem of transferring the speed of a vehicle between two values in a fixed interval of time using a predefined sequence of gears, and minimizing a cost related to fuel consumption, is solved in this paper. This is a hybrid dynamic optimization problem since the control variables include both a continuous variable (fuel flow) and a discrete variable (the gear to apply at each instant). The solution is given in the form of a hybrid optimal control algorithm that computes the optimal switching times between gears using Dynamic Programming and the optimal fuel profile between successive gear boundaries using a gradient algorithm to approximate the optimum conditions. In order to reduce the search of the optimal switching times to a search in a finite dimension graph, a procedure based on a changing grid is used. The algorithm is illustrated by a simulation using a diesel one-dimensional car model.

Keywords: Hybrid Optimization, Cruise Control, Optimal Control, Automotive.

1. INTRODUCTION

Both economic and environmental concerns are boosting the research in vehicle cruise control for both minimum fuel consumption and to reduce pollutant emissions Hashimoto *et al.* (2006). Improving the performance in this respect is increasingly seen by car manufacturers as a competitive advantage. This and other factors such as security, together with major progress in sensors and reliability of automotive electronics is boosting the incorporation of sophisticated control systems in cars, with the volume of software incorporated in the design growing exponentially. Depending on the type of system considered, cruise control provides help to the driver in selecting the optimal fuel flow for manoeuvres that may range from vehicle speed management or to move from the current point to the target destination using GPS and terrain map information. While some of the simplest functions may be currently found in relatively unexpensive cars, the area is the subject of research.

Recent papers address cruise control using optimal control, Dynamic Programming or Predictive Control techniques Gausemeier *et al.* (2010); Kolmanovsky and Filev (2010); Luu *et al.* (2010); Saerens *et al.* (2010). Cruise control for speed transfer is a hybrid optimization problem since the manipulated variables are both continuous (fuel flow) and discrete (gear ratio). In the last decade progress was made in methods for solving this type of problems, but the resulting algorithms usually require a high computational load Bemporad and Morari (2010); Bemporad and Giorgetti (2006); Hedlund and Rantzer (1999). As such, works on cruise control rely mostly on heuristic methods.

The contribution of this paper consists of a dynamic optimization based procedure to determine a suboptimal solution to the optimal switched systems state transfer problem, in a fixed interval of time, with given plant dynamics sequence. The procedure is applied to the cruise control speed transfer problem that minimizes a cost related to fuel consumption and using a predefined sequence of gears.

Where the explicit computation of the optimal control is needed, it is obtained by applying a recursive numerical gradient algorithm that provides an approximation to the conditions provided by Pontryagin's Optimum Principle.

The paper is organized as follows: After this introduction that motivates the problem, briefly reviews the relevant references and states the paper contribution and organization, the car model used is described in section 2. Section 3 describes the hybrid dynamic optimization procedure and section 4 provides a simulation example in which its convergence is illustrated. Finally, section 5 draws conclusions. For the sake of completeness a gradient algorithm to approximate the solution of the optimal control problem with terminal constraints is described in the appendix. This is one of the building blocks of the hybrid dynamic optimization proposed.

⋆ This work has been performed within the framework of *activity 2.4.1 – Smart drive control* of project *SE2A - Nanoelectronics for Safe, Fuel Efficient and Environment Friendly Automotive Solutions*, ENIAC initiative.

2. ONE-DIMENSIONAL NONLINEAR CAR MODEL

This section describes a one-dimensional model for a diesel car, with the following inputs:

- fuel flow as controlled input $[L/s]$;
- selected gear (manual gearbox is assumed);
- terrain inclination $[rad]$ and wind speed $[m/s]$ as disturbances.

The main output of the model is the car speed. Other quantities available from this model are:

(1) engine rotational speed $[rad/s]$;
(2) engine torque $[Nm]$;
(3) engine power $[kW]$;
(4) fuel consumption $[L/100km]$.

The dynamic model is build from elementary physical principles using information publicly available for a Toyota Avensis 2.0 D-4D SW for a 2007 model. All physical quantities are measured in SI units. Table 1 lists the values used for model parameters.

2.1 *Engine model*

The engine model assumes an input diesel flow $u(t)$ measured in liters per second. The total power P is given by

$$P(t) = Eu(t). \tag{1}$$

where E is the total energy density of diesel fuel. A considerable percentage of this power is dissipated in thermal losses, and only a part is available as mechanical power, P_m, given by

$$P_m(t) = \eta\big(T_e(t), w_e(t)\big) \times P(t). \tag{2}$$

where T_e and ω_e are the engine torque and speed and η is the efficiency. The engine torque output is given by

$$T_e(t) = \frac{P_m(t)}{\omega_e(t)} = \frac{\eta\big(T_e(t), w_e(t)\big) \times P(t)}{\omega_e(t)} \tag{3}$$

Equation (3) constitutes an algebraic loop, since efficiency η and engine torque values are computed based on each other, which makes computations more taxing. To overcome this, the torque value as a function of $w_e(t)$ and $u(t)$ can be numerically computed, by solving the algebraic loop (3) offline.

For the purposes of this work, it was assumed that efficiency level-curves on the (T_e, w_e) plane are elliptical (figure 1),

$$\eta(T_e, w_e) = \alpha - \beta \left[\frac{(T_e - c_T)^2}{l_T} + \frac{(w_e - c_w)^2}{l_w}\right] \tag{4}$$

for a reasonable choice of the parameters c_T, l_T, c_w and l_w. Constants α and β perform a linear transformation, making the elliptic surface concavity face downwards instead of upwards. Constant α is the value of the maximum efficiency, *i.e.* when $(T_e, w_e) = (c_T, c_w)$ then $\eta = \alpha$.

In this specific case a closed-form solution for computing $T_e(t)$ can be easily derived by replacing (4) in (3).

From the data available for this engine, it is known that it achieves a maximum torque of 310 Nm at 1800-2400 rpm. Below and above this operational range the torque is reduced. It is also known that a maximum power of 93kW is attained at 3600 rpm, implying a torque $T = \frac{93\times10^3}{3600}\frac{60}{2\pi} \approx 246.7$ Nm at that speed. From this scarce data, a maximum torque curve was designed as shown in figure 1. For any given engine

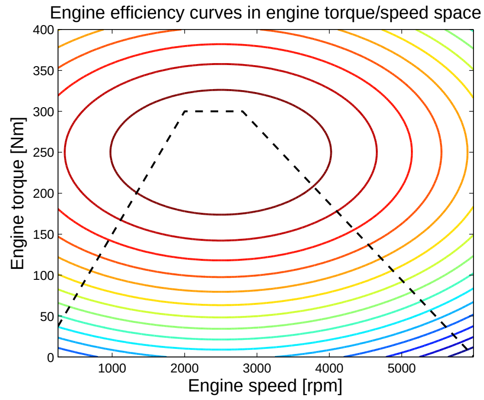

Fig. 1. Efficiency level-curves with maximum torque curve.

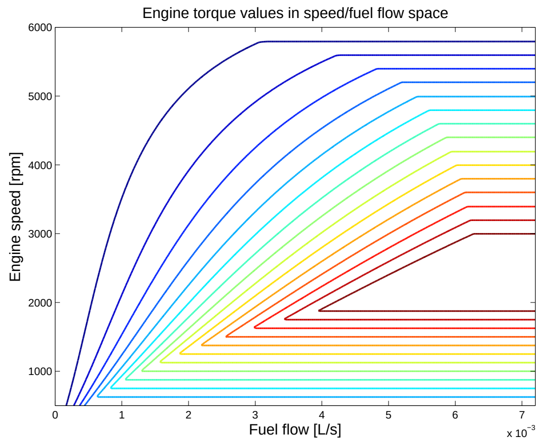

Fig. 2. Torque level-curves in engine speed/fuel flow space.

speed, admissible engine torque values lie below this curve. The numerical solution of equation (3) given the efficiency function (4) is represented in figure 2.

2.2 *Transmission*

The transmission links the wheels and the engine together using a gear box. Its role is to increase torque and decrease wheel speed to match the operational range of the engine. The transmission also introduces internal drag that depends on the engine speed. In the model developed here, the internal drag does not only model the transmission itself, but also all the load at the engine shaft.

The torque output available at the car wheels is given by

$$T_w(t) = r_i r_f \big(T_e(t) - \alpha_T - \beta_T \omega_e(t)\big) \tag{5}$$

where r_i is the gear ratio for gear i, r_f is the final drive ratio and α_T, β_T are drag coefficients.

Engine rotational speed, measured in $[rad/s]$, is obtained from wheel speed by gear ratio conversion and is given by

$$\omega_e(t) = r_i r_f \omega_w(t) \tag{6}$$

2.3 *Traction force and wheels*

The wheels are modeled as a rotational to linear movement converter neglecting inertia and drag. Wheel rotational speed, measured in $[rad/s]$, is given by

$$\omega_w(t) = \frac{2\pi}{\Lambda} v(t) \tag{7}$$

parameter	value	units
m	1500	Kg
ρ	1.2	kg/m^3
A	2.29	m^2
C_d	0.29	-
Λ	1.9852	m
α_T	35	Nm
β_T	0.07	$Nm/rad.s^{-1}$
E	$40.8e6$	J/L

Table 1. Non-linear car model parameter values

gear ratio	value
r_1	3.818
r_2	1.913
r_3	1.218
r_4	0.860
r_5	0.790
r_6	0.673
r_f	3.240

Table 2. Non-linear car model gear ratio values

where Λ is the wheel perimeter. The used tire dimensions are 205/55R16, corresponding to a perimeter of $\Lambda = 1.9852$ meters.

Similarly, the traction force is obtained from the torque applied by the engine at the wheels

$$F(t) = \frac{2\pi}{\Lambda} T_w(t) \tag{8}$$

2.4 *Car dynamic model*

The evolution of the car speed depends of the forces applied. The forces considered are: traction force $F(t)$, gravitational force and aerodynamic drag $F_a(t)$.

$$\dot{v}(t) = -9.8\sin(\theta) + \frac{1}{m}\left(F(t) - F_a(t)\right) \tag{9}$$

where θ is the terrain inclination. Aerodynamic drag is assumed to be given by

$$F_a(t) = \frac{1}{2}\rho A C_d \big(v(t) - v_{wind}(t)\big)^2 \tag{10}$$

where ρ is the air density, A is the frontal area of the vehicle, C_d is the drag coefficient, and m is the car mass.

Table 1 contains the values used for the nonlinear car model parameters. For the purposes of modeling the transmission, a six-speed manual gear box is used. Table 2 shows the published gear ratios of the gear box along with the final drive ratio.

3. HYBRID DYNAMIC OPTIMIZATION ALGORITHM

The car dynamic model (9) may be represented by the nonlinear state-space model

$$\dot{x}(t) = f_i(x(t), u(t), t), \quad i = 1, ..., N \tag{11}$$

where the state, given by the car speed, is scalar and the f_i are vector fields corresponding to each of the gears i, for $i = 1, ..., N$ where N is the number of gears.

The following procedure consists of an algorithm to find a suboptimal solution for the optimal switched dynamics state transfer problem, *i.e.* making the state change from an initial value x_0 at t_0 to a desired target value x_f at the final time t_f, where t_0 and t_f are given and fixed. The algorithm is based on dynamic optimization in that the desired state transfer, x_0 to x_f, is broken up into smaller state transfer problems, one for each distinct dynamics. The sequence of dynamics equations and the state values at which a switch between dynamics equations occur are given. The performance index, to minimize has the form

$$J = \int_{t_0}^{t_f} L(x(t), u(t))dt \tag{12}$$

where $x(t)$ and $u(t)$ are the state and control vectors at some time t, respectively. In the case considered, the lagrangian is given by $L = u^2(t)/2$. Let

$$X_i \quad i = 1, ..., N+1 \tag{13}$$

be the state values at which a switch between plant dynamics occurs, which are given and fixed, such that the plant dynamics f_i apply when performing state value transfers from X_i to X_{i+1}. The initial and final state values are x_0 and x_f. The dynamic optimization procedure consists of finding the optimal switching instants

$$t_i^*, i = 1, ..., N+1 \tag{14}$$

in respect to the chosen state values X_i, where t_i^* corresponds to the time instant at which the state value X_i occurs. Naturally, the first and last time instants are $t_1^* \equiv t_0$ and $t_{N+1}^* \equiv t_f$. In order to determine the remaining $N-1$ optimal switching instants, a set of M_i candidate time values is defined

$$t_i^j, \quad j = 1, ..., M_i \tag{15}$$

at which each of the $N-1$ intermediate state values $X_i, i = 2, ..., N$ are attained.

This defines a grid of state/time values that can also be represented as an oriented graph, as shown in figure 3. The cost of each arc of the graph is computed by (12), where the integration interval corresponds to the time interval where that arc is defined. The optimal switching instants (in respect to all the candidates) are computed as those for which the total cost for transferring the state value from x_0 at t_0 to x_f at t_f is minimum. Having computed all the $N+1$ optimal switching instants, the resulting set of instants can be adjusted and refined by creating new sets of candidate time values (15) and repeating the same procedure.

In the sequel, when referring to t_i^j, we will use the notion of time value and node interchangeably, since the lower index i already indicates that this is a time value belonging to state value X_i.

The total procedure can be separated into three phases:

1. *Define graph nodes and arcs*

Nodes: The initial and final state values correspond to two nodes, t_0 and t_f, respectively. For the remaining state values, $X_2, ..., X_N$, define a set of candidate nodes (15), such that $t_i^1 < t_{i+i}^1$ and $t_i^{M_i} < t_{i+1}^{M_{i+1}}$, for $i = 2, ..., N$, *i.e.* the first and last candidate nodes of switch i must be smaller than the first and last candidate nodes of switch $i+1$ (see figure 3).

Arcs: Acceptable arcs on the graph are all those that connect a node at switch X_i with another node at the following switch X_{i+1}, and such that the time of the former node is strictly smaller than the time of the latter.

2. *Compute optimal control, and corresponding cost, for all arcs in graph*

For each arc connecting nodes t_i^j and t_{i+1}^k, compute the optimal control signal $u_{i,j}^{i+1,k}$ using the algorithm described in Appendix

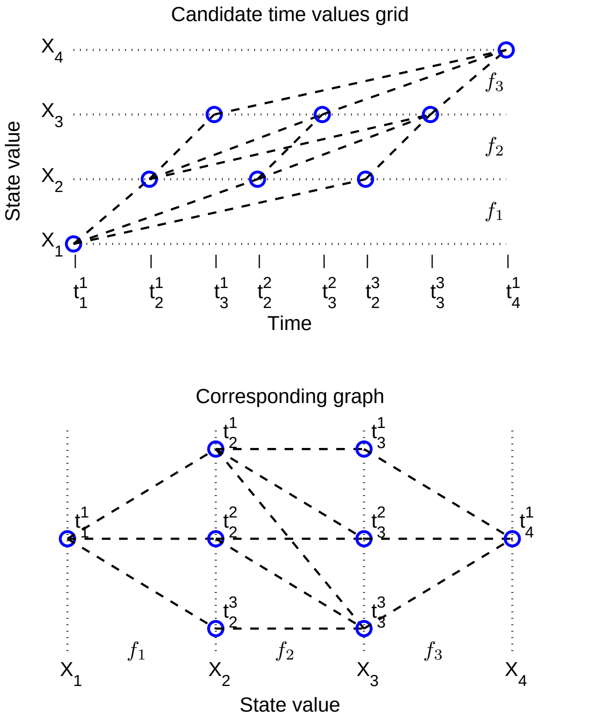

Fig. 3. Candidate time values for a switched dynamics optimization problem with 3 plants, f_1, f_2 and f_3

Bryson and Ho (1975), and the corresponding performance index value $J_{i,j}^{i+1,k}$, for performing the optimal state value transfer from X_i to X_{i+1}. If a given arc is impossible, *e.g.* the corresponding state transfer cannot be accomplished in the time interval that spans between t_i^j and t_{i+1}^k, the cost of that arc is set to $+\infty$.

3. *Determine optimal path and corresponding cost*

The minimum-cost path that connects the first and last nodes, t_0 and t_f, is the optimal path with respect to the candidate nodes established in *1*. The nodes that make up the optimal path correspond to the optimal switching instants (14). To compute the optimal path, a dynamic programming approach is defined as follows. For any given node t_i^j there is an optimal path that connects that node to the final node t_f. Let $u_{t_i^j}^*$ be the optimal control signal that connects t_i^j to the next node on its optimal path and $J_{t_i^j}^*$ the cost of the whole optimal path connecting t_i^j to t_f. The dynamic programming approach consists of going through all state values backwards, from X_N to X_1, and for each node solving

$$u_{t_i^j}^* = \underset{u_{i,j}^{i+1,k}}{argmin} \left\{ J_{i,j}^{i+1,k} + J_{t_{i+1}^k}^* \right\}. \tag{16}$$

where the first term inside the brackets corresponds to the cost of going from t_i^j to t_{i+1}^k and the second term corresponds to the optimal cost of going from t_{i+1}^k to the final node t_f. The optimal cost of the node, $J_{t_i^j}^*$, is then computed as the sum of both terms inside the brackets when (16) is achieved. The optimal decision for node t_i^j, *i.e.* the node t_{i+1}^k for which (16) is achieved, is also registered.

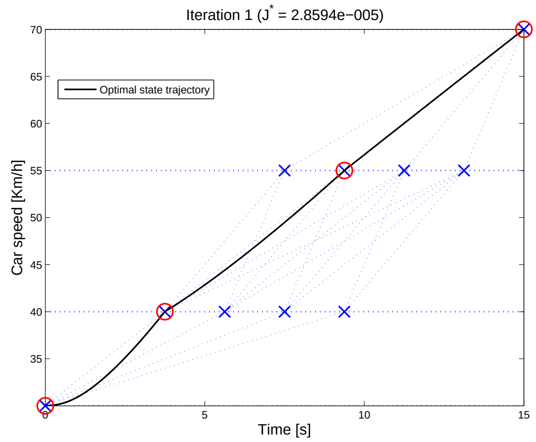

Fig. 4. Candidate time values and optimal path after iteration 1

When all nodes on the graph have been labeled with the optimal cost to go an the optimal decision, the total optimal cost for the state transfer problem is $J^* \equiv J_{t_0}^*$. The optimal path can be easily reconstructed by starting from the first node t_0 and recursively jumping to the optimal decision until the final node t_f is reached. Finally, the total optimal control signal u^* can be recovered from the individual optimal control signals along the optimal path.

It is remarked that the solution is suboptimal in the sense that the switching state values X_i are fixed and that the global optimum may not be found.

Having determined the optimal nodes, *i.e.* the optimal switching instants (14), the process can be repeated by going back to *1* and choosing new candidate nodes nearer to the optimal ones computed in this iteration.

4. RESULTS FOR THE SPEED TRANSFER PROBLEM WITH 3 GEARS

In this section, the hybrid dynamic optimization algorithm is applied to solve the minimum energy speed transfer (MEST) problem for the non-linear car model, using a predefined set of 3 gears. Here, the MEST problem was considered with $t_0 = 0$ and $t_f = 15$ seconds, with initial and final car speeds $v_0 = 30$ Km/s and $v_f = 70$ Km/h, such that gear 2 is used when $30 \leq v < 40$, gear 3 is used when $40 \leq v < 55$ and gear 4 is used when $55 \leq v < 70$. Terrain inclination and wind speed were assumed to be null.

After 14 iterations, the resulting optimal switching instants are $t_2^* = 2.662$ seconds and $t_3^* = 8.049$ seconds with a corresponding performance index value of $J^* = 28.2 \times 10^{-6}$. The evolution of the performance index along iterations of the hybrid dynamic optimization algorithm is shown in figure 9. The resulting optimal control signal is shown in figure 10.

5. CONCLUSIONS

A hybrid dynamic optimization algorithm for solving the minimum energy speed transfer problem in cruise control is presented. The algorithm proposed relies on a changing grid of possible instants to switch between gears and combines Dynamic Programming with Constrained Optimal Control. The algorithm is formulated for first order hybrid systems but may be extended to higher orders at the expense of computational

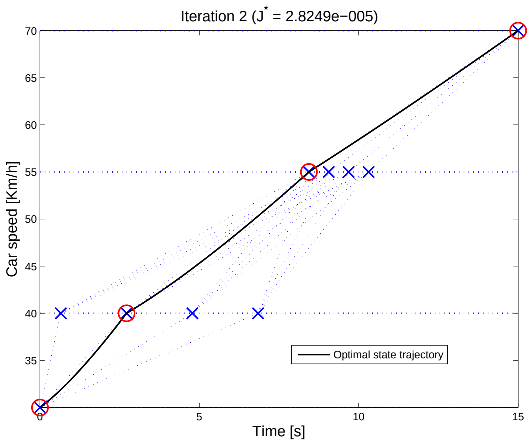

Fig. 5. Candidate time values and optimal path after iteration 2

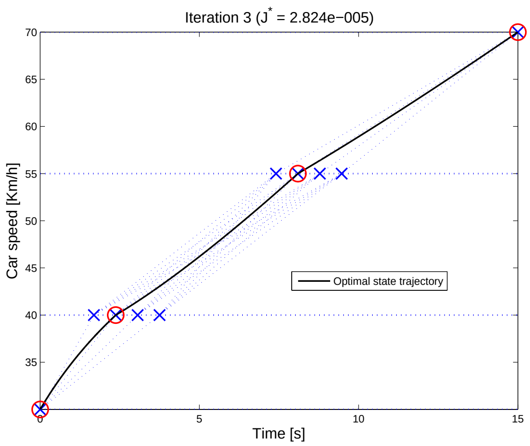

Fig. 6. Candidate time values and optimal path after iteration 3

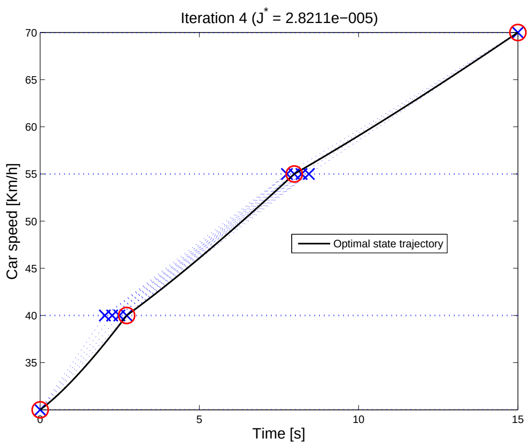

Fig. 7. Candidate time values and optimal path after iteration 4

load. In turn, this may be reduced by replacing the gradient algorithm to solve state transfer with constant dynamics by more efficient algorithms Ferreau *et al.* (2006).

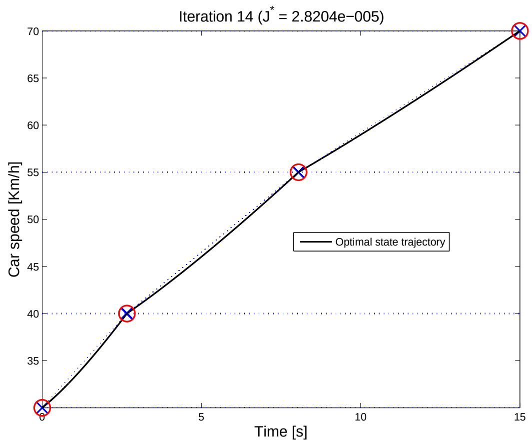

Fig. 8. Candidate time values and optimal path after iteration 14

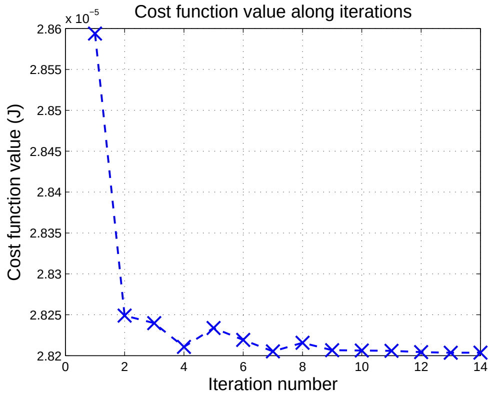

Fig. 9. Evolution of the performance index value along 9 iterations of the hybrid dynamic optimization procedure

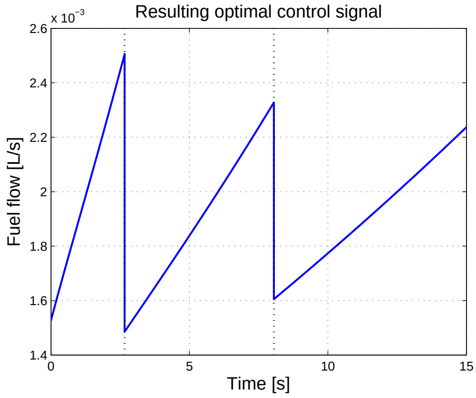

Fig. 10. Resulting optimal control for the speed transfer problem.

REFERENCES

Bemporad, A. and N. Giorgetti (2006). Logic-Based Solution Methods for Optimal Control of Hybrid Systems. *IEEE Trans. Autom. Control*, 51(6):963-976.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics and constraints. *Automatica*, 35:407-427.

Bryson, A. E., and Ho, Y. C. (1975). *Applied Optimal Control*.

Ferreau, H. J., Lorini, G. and Diehl, M. (2006). Fast nonlinear model predictive control of gasoline engines. *Proc. 2006 IEEE Int. Conf. on Control Applications*, Munich, Germany, 2754-2759.

Gausemeir, S., K.-P. Jaker and A. Trachtler (2010). Multi-objective Optimization of a Vehicle Velocity Profile by

Means of Dynamic Programming. *Prep. IFAC Symp. Advances in Automotive Control, AAC2010*, Munich, Germany.

Hashimoto, S. , Okuda, H., Okada, Y., Adachi, S., Niwa, S. and Kajitani, M. (2006). An engine control systems design for low emission vehicles by generalised predictive control based on identified model. *Proc. 2006 IEEE Int. Conf. on Control Applications*, Munich, Germany, 2411-2416.

Hedlund, S. and A. Rantzer (1999). Optimal Control of Hybrid Systems. *Proc. 38th CDC*, Phoenix, Arizona, 3972-3977.

Kolmanovsky, I. V. and D. P. Filev (2010). Terrain and traffic Optimized Vehicle Speed Control. *Prep. IFAC Symp. Advances in Automotive Control, AAC2010*, Munich, Germany.

Luu, H. T., L.Nouvelire and S. Mammar (2010). Dynamic Programming for fuel consumption optimization on light vehicle. *Prep. IFAC Symp. Advances in Automotive Control, AAC2010*, Munich, Germany.

Saerens, B., J. Vandersteen, T. Persoons, M. Diehl and E. Van den Bulck (2009). Minimization of the fuel consumption of a gasoline engine using dynamic optimization. *Applied Energy* 86:1582-1588.

Appendix A. ITERATIVE SOLUTION FOR THE OPTIMAL CONTROL PROBLEM WITH TERMINAL CONSTRAINTS

Let

$$\dot{x} = f(x, u, t), \qquad t \in [t_0, t_f], \quad x(t_0) = x_0 \tag{A.1}$$

describe the nonlinear time-varying dynamics of a plant, with given initial condition x_0, where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ are the state and input vectors at time t, respectively, and

$$J = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt \tag{A.2}$$

a performance index to maximize. A minimization problem can also be formulated by maximizing the performance index $J_{min} = -J$.

The optimal control problem with terminal constraints consists in finding the input signal $u^*(t)$, $t \in [t_0, t_f]$, for which the plant exhibits a state trajectory $x^*(t)$ such that the terminal state value, $x^*(t_f)$, verifies a set of terminal constraint functions, in the form of q restriction equations,

$$\psi(x(t_f), t_f) = \begin{bmatrix} \psi_1(x(t_f), t_f) \\ \cdots \\ \psi_q(x(t_f), t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ \cdots \\ 0 \end{bmatrix} \tag{A.3}$$

and such that the performance index value, J, is maximum.

An iterative numerical algorithm for obtaining the optimal control signal under these circumstances is now described Bryson and Ho (1975). An initial estimate for the optimal control signal must be provided. Each iteration consists of the following steps:

1. *Integrate state equation*

Using the current estimate of the optimal control signal, $u(t) \in \mathbb{R}^m$, integrate the state equation A.1 to obtain the state evolution $x(t) \in \mathbb{R}^n$ from t_0 to t_f.

2. *Integrate co-state equations*

Let $\lambda^\Phi(t)$, an $(n \times 1)$ vector, and $\lambda^\Psi(t)$, an $(n \times q)$ matrix, be co-state variables. Define

$$H^\Phi(\lambda^\Phi, x, u, t) = L(x, u, t) + {\lambda^\Phi}^T f(x, u, t),$$
$$H^\Psi(\lambda^\Psi, x, u, t) = {\lambda^\Psi}^T f(x, u, t),$$

and integrate backwards, from t_f to t_0, the adjoint equations

$$-[\dot{\lambda}^\Phi]^T = H_x^\Phi = \frac{\partial L}{\partial x} + {\lambda^\Phi}^T \frac{\partial f}{\partial x},$$
$$-[\dot{\lambda}^\Psi]^T = H_x^\Psi = {\lambda^\Psi}^T \frac{\partial f}{\partial x},$$

for which the terminal co-state conditions are

$$\lambda^\Phi(t_f) = \phi_x^T(x(t_f), t_f),$$
$$\lambda^\Psi(t_f) = \psi_x^T(x(t_f), t_f).$$

3. *Compute Hamiltonian partial derivatives*

Compute the Hamiltonian functions partial derivatives with respect to the control signal u for all $t \in [t_0, t_f]$,

$$H_u^\Phi = \frac{\partial L}{\partial u} + {\lambda^\Phi}^T \frac{\partial f}{\partial u}$$
$$H_u^\Psi = {\lambda^\Psi}^T \frac{\partial f}{\partial u}$$

$H_u^\Phi(t)$ is a $(1 \times m)$ vector and $H_u^\Psi(t)$ is a $(q \times m)$ matrix.

4. *Compute Lagrange multiplier vector* ν

Compute ν ($q \times 1$ vector)

$$\nu = -Q^{-1} g$$

where

$$g = \int_{t_0}^{t_f} H_u^\Psi(t)[H_u^\Phi(t)]^T dt$$

is a $(q \times 1)$ vector and

$$Q = \int_{t_0}^{t_f} H_u^\Psi(t)[H_u^\Psi(t)]^T dt$$

is a $(q \times q)$ matrix.

5. *Compute gradient signals*

Evaluate ψ at the terminal time and compute the gradient signals $\delta u_k(t)$ and $\delta u_\eta(t)$ for all $t \in [t_0, t_f]$

$$\delta u_k(t) = [H_u^\Phi(t) + \nu^T H_u^\Psi(t)]^T$$
$$\delta u_\eta(t) = [H_u^\Psi(t)]^T Q^{-1} \psi(t_f)$$

6. *Compute and update the estimate of the optimal control signal*

Compute the control correction signal $\delta u(t)$

$$\delta u(t) = -k\delta u_k(t) - \eta \delta u_\eta(t)$$

and update the estimate of the optimal control signal

$$u(t) \leftarrow u(t) + \delta u(t)$$

choosing $k < 0$ ($k > 0$) if maximizing (minimizing) the performance index, and $0 < \eta \leq 1$.

7. *Evaluate stop criteria*

Compute the root-mean-square value of $\delta u_k(t)$ and $\delta u_\eta(t)$ using the standard definition

$$(x)_{rms} = \sqrt{\frac{1}{t_f - t_0} \int_{t_0}^{t_f} [x(t)]^2 dt}$$

The algorithm stops if $max\{(\delta u_k)_{rms}, (\delta u_\eta)_{rms}\}$ is smaller than a specified threshold, or if the maximum number of iterations is reached.