Hybrid Control of an Automotive Robotized Gearbox for Reduction of Consumptions and Emissions

Alberto Bemporad[1], Pandeli Borodani[2], and Massimo Mannelli[1]

[1] Dip. Ingegneria dell'Informazione
Università di Siena
Via Roma 56, 53100 Siena, Italy
bemporad@dii.unisi.it, massimo.mannelli@PSD.Invensys.com
[2] FIAT Research Center
Vehicle Control Systems
Strada Torino 50 , 10043 Orbassano(TO), Italy
p.borodani@crf.it

Abstract. This paper describes the application of hybrid modeling and receding horizon optimal control techniques for supervising an automotive robotized gearbox, with the goal of reducing consumptions and emissions, a problem that is currently under investigation at Fiat Research Center (CRF). We show that the dynamic behavior of the vehicle can be easily approximated and captured by the hybrid model, and through simulations on standard speed patterns that a good closed loop performance can be achieved. The synthesized control law can be implemented on automotive hardware as a piecewise affine function of the measured and estimated quantities.

1 Introduction

The automotive market analysts forecast for the automatic transmission system a relevant growth in the near future [1]. Conventional automatic transmissions provide a good level of comfort, but evidence significant drawbacks concerning other aspects: fuel economy, cost, weight, and size. Also, many customers (particularly in European countries) associate with the manual gearbox a significant value in terms of driving feeling and expectation.

Recent technological developments try to satisfy the increasing demand of the automotive market for automatic transmissions, matching at the same time the conflicting requirements of comfort, performance, fuel economy and cost reduction. An extremely promising system is the automated gearbox, named in this paper "robotized gearbox" (Fig. 1); it is based on servo actuators applied to a standard mechanical gearbox. The new transmission system with both automatic and semiautomatic operating modes [2,3,4,5] is directly derived from a standard manual gearbox by adding electronically controlled small servo-hydraulic actuators, capable to move better than the human driver the gearshift mechanisms [6].

O. Maler and A. Pnueli (Eds.): HSCC 2003, LNCS 2623, pp. 81–96, 2003.

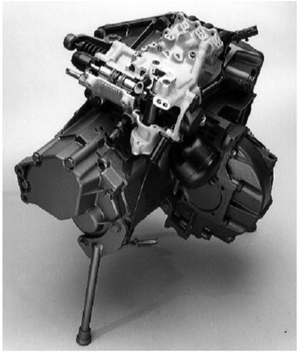


Fig. 1. Robotized gearbox

The introduction of the robotized gearbox gives the opportunity to transfer the driver's request to a higher level and to use automatic criteria for optimizing the lower level. An automatic system supervisor can in fact control the gear shifting and the torque regulators with the duty of choosing the gear and the engine torque, satisfying the requests of the driver, the constraints, and optimizing the powertrain behavior, reducing consumptions and emissions. Emissions are of particular interest, as in recent years the European Community has stressed the noxious effects of emissions, and is trying to drastically reduce them in the near future.

In this paper we show how the whole system (vehicle and robotized gearbox) can be modeled as a hybrid one in order to synthesize a supervisor that brings the engine torque close to the Optimal Operating Line (OOL), while minimizing consumptions and emissions. The system is indeed intrinsically hybrid, as once the gear (a discrete input) is selected, a different continuous dynamics results.

Current CRF control strategies are mainly based on static maps, as in most automatic gear shifting schemes nowadays in production. Such control schemes are motivated by the fact that the system is nonlinear. The presence of nonlinearities and constraints on one hand, and the simplicity needed for real-time implementation on the other, have discouraged the design of optimal control strategies for this kind of problem. Recently, a new framework for modeling hybrid systems was proposed in [7], and an algorithm for synthesizing piecewise affine optimal controllers for such systems in [8]. In this paper we describe how the hybrid framework [7] and the optimization-based control strategy [8] can be successfully applied for solving this problem in a systematic way. More in detail, for solving the gearbox control problem we need to design a supervisor (depicted as MPC Controller in Fig. 2) that in real-time decides the best gear that minimizes consumptions and emissions and, at the same time, guarantees a good tracking of the desired traction power. As these are conflicting objectives, besides the gear the supervisor is allowed a second degree of freedom, namely to deviate the desired requested engine torque $T_E(\omega_{engine})$ by a quantity Δ_{Torque}. The idea is to solve the posed control problem by formulating a model-based receding

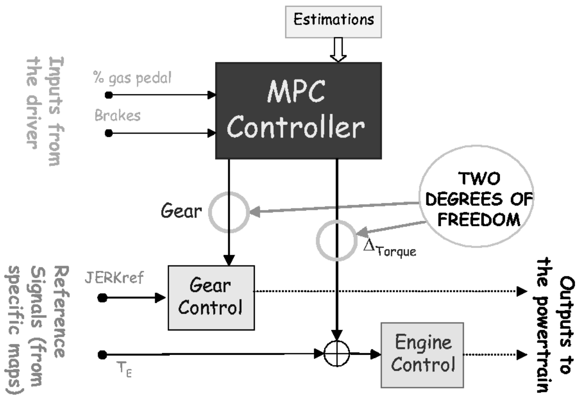


Fig. 2. Hybrid supervisor for a robotized gearbox with two degrees of freedom

horizon optimal control problem which minimizes consumptions, emissions, and the deviation Δ_{torque} from the desired torque. We show, through simulations on a simplified model and for a set of parameters provided by CRF, that good performances can be achieved, particularly comparing our results with the ones obtained by CRF and with the ones provided by the European Union on standard speed patterns. Furthermore, the resulting optimal controller consists of a piecewise affine function of the measurements, that can be easily implemented.

2 Vehicle Model

With the objective in mind of controller design, the vehicle model considered here is highly simplified, although it still allows the synthesis of a reasonably performing control action, as will be shown in Section 5. The model consists of the equations [9, 10]

$$\dot{\omega}_{sec} = \frac{1}{J_P(\tau_c)}\left(T_E\tau_c - \beta_2\omega_{sec} - \frac{2}{\tau_p}T_{axle}\right) \tag{1a}$$

$$\dot{V}_X = \frac{1}{m}\left(\frac{2}{R_e}(T_{axle} - T_{brake} - T_{rot}) - F_{friction} + F_{slope}\right) \tag{1b}$$

$$\dot{T}_{axle} = K_{sa}\left(\frac{\omega_{sec}}{\tau_p} - \frac{V_X}{R_e}\right) + \beta_1\frac{d\left(\frac{\omega_{sec}}{\tau_p} - \frac{V_X}{R_e}\right)}{dt} \tag{1c}$$

with

$$F_{friction} = \frac{1}{2}\rho\, V_X^2\, S\, C_x, \tag{1d}$$

Table 1. Physical quantities and parameters of the vehicle model

Name	Description	
ω_{sec}	Secondary shaft speed	rpm
V_X	Vehicle velocity	m/s
T_{axle}	Torque about the axle shaft	Nm
T_E	Engine torque	Nm
τ_c	Gear ratio (when gear engaged)	
τ_p	Bridge ratio	
T_{brake}	Brake torque	Nm
$J_P(\tau_c)$	Equivalent primary inertia	kgm^2
m	Vehicle mass	m
R_e	Rolling wheel radius	m
T_{rot}	Rolling resistance torque	Nm
F_{slope}	Gravity contribution due to roadway slope	N
β_1	Axle coefficient	kgm^2/s
K_{sa}	Axle coefficient	kgm^2/s^2
β_2	Combustion dynamic coefficient	kgm^2/s
ρ	Air density	kg/m^3
S	Frontal area of the vehicle	m^2
C_x	Aerodynamic drag coefficient	
$JERK_{ref}$	Reference jerk during gear shifting	m/s^3

where the involved physical quantities and parameters are described in Table 1. The first equation represents the engine dynamics, the second one describes the longitudinal motion dynamics of the vehicle, the third equation is referred to the axle dynamics. The friction force is approximated as a linear function of the velocity, based on a best fit on the range [15, 120] km/h, which is the range where the gear is most often shifted. The rolling resistance torque is approximated as constant, as this force, compared to the other friction forces, has no meaningful variations. Terms like $J_P(\tau_c)$ show that those parameters depend on the gear ratio τ_c, as there are indeed five different linear dynamics, one for each gear. Fig. 3 shows the position of the open loop poles for each one of them. Moreover, we assume that the requested engine torque is immediately applied (therefore neglecting the delay due to the torque control loop) and that it corresponds to the actual torque delivered by the engine.

3 Hybrid Model

Hybrid systems provide a unified framework for describing processes evolving according to continuous dynamics, discrete dynamics, and logic rules [11,12,13,14]. The interest in hybrid systems is mainly motivated by the large variety of practical situations, for instance embedded control systems, where physical processes interact with digital controllers. Several modeling formalisms were developed by various researchers to describe hybrid systems, among them the class of Mixed

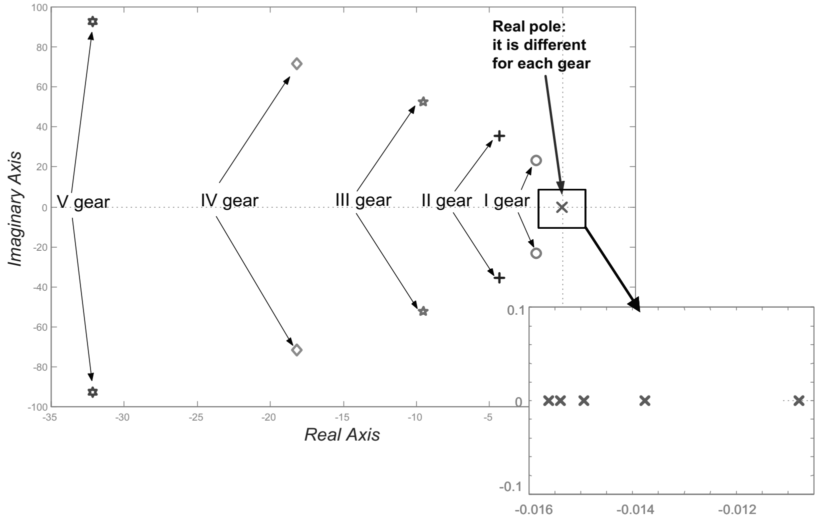


Fig. 3. Position of the poles for each gear

Logical Dynamical (MLD) systems introduced in [7]. Examples of real-world applications that can be naturally modelled within the MLD framework are listed in [15], where the authors describe the language HYSDEL (Hybrid System Description Language) for obtaining an MLD model from a high level textual description of the hybrid dynamics. HYSDEL is used here to “hybridize” the vehicle model (1), as reported in Appendix A. Such a model is obtained through the following steps:

– Discretize the model with sampling time $T_s = 0.3$ s. This value corresponds to the average synchronization time of the robotized gearbox.
– Introduce a Boolean input $gear^i \in \{0, 1\}$ for each gear $i = 1, \dots, 4$, with $\delta^i = 1$ if and only if the corresponding gear $\#i$ is engaged. The condition “gear #5 engaged” is then represented by $gear^1 = gear^2 = gear^3 = gear^4 = 0$.
– Introduce an auxiliary continuous variable $\omega_{sec}(j)$ for each gear $\#j$, $j = 1, \dots, 5$, and set $\omega_{sec} = \sum_{j=1}^{5} \omega_{sec}(j)$, where only one variable $\omega_{sec}(j)$ is nonzero at a time.
– Reduce the order of the linear dynamics (1b) [1] and get a model with only one state, in order to simplify the control algorithm.
– Add the following constraints in order to guarantee the correct operation of the engine:

[1] Order reduction is achieved by first obtaining a balanced realization using the MATLAB® function `balreal`, and then by reducing the order using the MATLAB® function `modred`.

- On the primary shaft speed, ω_{engine} must be in the range $[700, 6000]$ rpm. This requires a constraint on each secondary shaft speed of the form $\omega(j)_{min} < \omega(j)_{sec} < \omega(j)_{max}$, where $j = 1, \dots, 5$ is the gear.
- On the two manipulated variables: the variation Δ_{Torque} from the nominal engine torque $T_E(\omega_{engine})$ is constrained in the range $\Delta_{Torque,min} \leq \Delta_{Torque} \leq \Delta_{Torque,max}$, while concerning the gear, we have the constraint that only one gear can be selected at a time.
- On the braking torque that can be directly applied, the range is [0, 1150] Nm.
- On the engine torque, in order to avoid applying an excessive torque, $T_E(\omega_{engine}) + \Delta_{Torque} < T_{available}(\omega_{engine})$.

The above dynamic equations and constraints are modeled in HYSDEL, as reported in Appendix A, and translated by the HYSDEL compiler into the MLD form

$$x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \tag{2a}$$

$$y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \tag{2b}$$

$$E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4x(t) + E_5, \tag{2c}$$

where $x \in \mathbb{R}^4$, ($x_1 = x_{red}$, $x_2 = T_{ref}$, $x_3 = T_{brake}$, $x_4 =$ road slope, $u \in \mathbb{R} \times \{0,1\}^4$, ($u_1 = \Delta_{torque} \in \mathbb{R}$, $u_{j+1} = gear^j \in \{0,1\}$, $j = 1, \dots, 4$), $y \in \mathbb{R}^2$, ($y_1 = \omega_{engine}$, $y_2 = T_E$), $\delta \in \{0,1\}^2$ and $z \in \mathbb{R}^{12}$. The state x_{red} is the state of the reduced-order model, the other states are actually just measured variables, and all the inputs are manipulated variables.

In order to validate the model, in Fig. 4 we compare the open-loop evolution of the discrete-time MLD model (2) and of the nonlinear continuous time model (1), under the same inputs (% of gas pedal and gear). It is apparent that the MLD model captures in discrete time the hybrid behavior of the system quite satisfactorily. It may be noted that there is a small offset due to the approximation of the friction term: this is not a problem, as the offset will be compensated by the feedback control action from actual measured values. The validity of the hybrid MLD model is also confirmed by the fact that the "ground power"[2] requested by both models is practically the same (Fig. 5).

4 Optimization-Based Control Design

We describe how receding horizon optimal control for hybrid systems [7,8] can be usefully employed here to design a control law for the robotized gearbox control problem. The main idea is to setup a finite-horizon optimal control problem for the hybrid MLD system by optimizing a performance index under constraints,

[2] The ground power is the total power that the vehicle receives from the external environment (the ground). Given the engine torque request T_E, it is computed using the following relations: $T_G = T_E\eta_t\tau_p\tau_c$ and $P_GR_e = T_GV_X$, where T_G, η_t, and P_G are respectively ground torque, engine efficiency, and ground power.

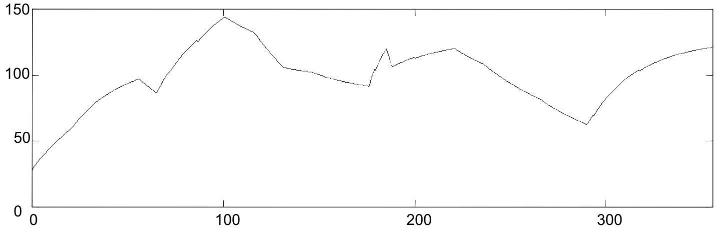


Fig. 4. Comparison of speed profiles: nonlinear model (1) vs. hybrid MLD model (2)

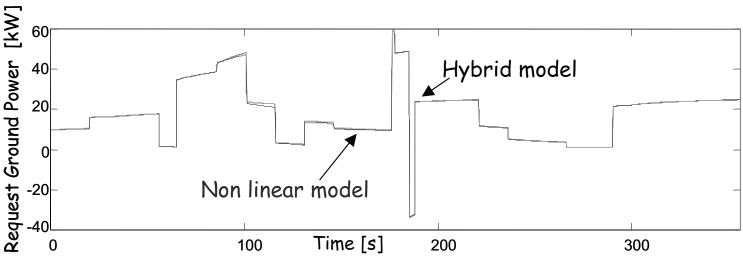


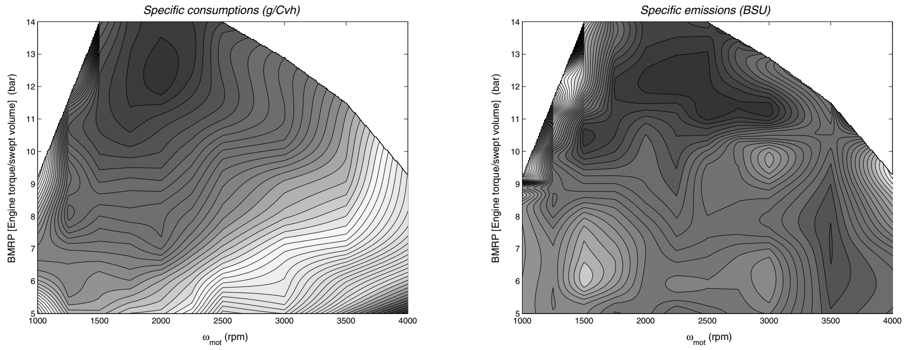
Fig. 5. Comparison of requested ground power: nonlinear model (1) vs. hybrid MLD model (2)

with the goal of minimizing consumptions and emissions and, at the same time, of guaranteeing a good tracking of the desired traction power.

The performance index we attempt at minimizing will contain a term that penalizes the input command Δ_{torque} (=deviation of the requested engine torque from the nominal one) and two functions $f_1(C)$ and $f_2(E)$ that express the value of consumptions and emissions, respectively. These functions, as shown in Fig. 6, are highly nonlinear. In order to use linear programming solvers, we need to approximate $f_1(C)$ and $f_2(E)$ as piecewise affine maps. With the goal of minimizing consumptions and emissions, the supervisor should bring the outputs of the MLD system (engine speed and engine torque[3]) as close as possible to the zone where the consumptions/emissions are lowest. As can be seen from Fig. 6(a), the zone of minimum consumption is located near the zone where the engine torque is maximum[4]. This means that minimizing consumptions does not necessary imply a lower efficiency. On the contrary, if the right gear is chosen, it is possible to maintain the same speed without loosing efficiency. We approximate function $f_1(C)$ ($f_2(E)$) as a piecewise affine function on the difference between the system outputs and the coordinates of the point y_{cons} of minimum

[3] In the maps the ordinate is expressed in BMRP: it represents the ratio between the engine torque and the swept volume; the second term depends on the volume of the engine, once the engine is engaged, it is a constant.

[4] The engine torque is maximum on the range [1500,2500] rpm; it is the same range where BMRP is maximum because of the relation $T_E = BMRP \cdot swept\ volume$



(a) Consumptions map

(b) Emissions map

Fig. 6. Consumptions and emissions maps for the specific examined engine. The darkest zones represent the zones where consumptions (emissions) are minimum

consumptions (y_{emiss} of minimum emissions). As an example, Fig. 7 shows the resulting piecewise affine approximation for $f_1(C)$.

The resulting finite-time hybrid optimal control problem is the following:

$$\min_{u_0^{T-1},\delta_0^{T-1},z_0^{T-1}} J(u_0^{T-1},x_0) \triangleq \sum_{k=0}^{T-1} \rho \cdot \|(\Delta_{torque}(t+k|t))\|_\infty \; +$$
$$\rho_c \cdot \left\| \begin{bmatrix} q_{c_1} & q_{c_2} \\ q_{c_3} & q_{c_4} \end{bmatrix} \left(\begin{bmatrix} y_1(t+k|t) \\ y_2(t+k|t) \end{bmatrix} - y_{cons} \right) \right\|_\infty + \; \rho_e \cdot \left\| \begin{bmatrix} q_{e_1} & q_{e_2} \\ q_{e_3} & q_{e_4} \end{bmatrix} \left(\begin{bmatrix} y_1(t+k|t) \\ y_2(t+k|t) \end{bmatrix} - y_{emiss} \right) \right\|_\infty \tag{3}$$

$$\text{subject to} \begin{cases} x_0 = x(t) \\ x_{k+1} = Ax_k + B_1 u_k + B_2 \gamma_k + B_3 z_k \\ y_k = Cx_k + D_1 u_k + D_2 \gamma_k + D_3 z_k \\ E_2 \gamma_k + E_3 z_k \leq E_1 u_k + E_4 x_k + E_5, \end{cases}$$

where $x(t)$ is the state of the MLD system at time t, and $\|\cdot\|_\infty$ is the standard ∞-norm. Matrices $Q_1 = \left[\begin{smallmatrix} q_{c_1} & q_{c_2} \\ q_{c_3} & q_{c_4} \end{smallmatrix}\right]$ and $Q_2 = \left[\begin{smallmatrix} q_{e_1} & q_{e_2} \\ q_{e_3} & q_{e_4} \end{smallmatrix}\right]$ are the weighting matrices needed for the approximated piecewise affine consumption and emission functions. By varying the weights ρ_c, ρ_e we are able to emphasize the reduction of consumptions or emissions.

In (3) we assume that possible physical and/or logical constraints on the variables of the hybrid system are already included in the mixed-integer linear constraints of the MLD model, as they can be conveniently modeled through the language HYSDEL. Receding horizon control (RHC) amounts to repeatedly computing the optimal solution to (3) at each time t, and applying only the first optimal control move u_0^* as the input $u(t)$ to the system. Problem (3) can be translated into a mixed integer linear problem (MILP), i.e., into the minimization of a

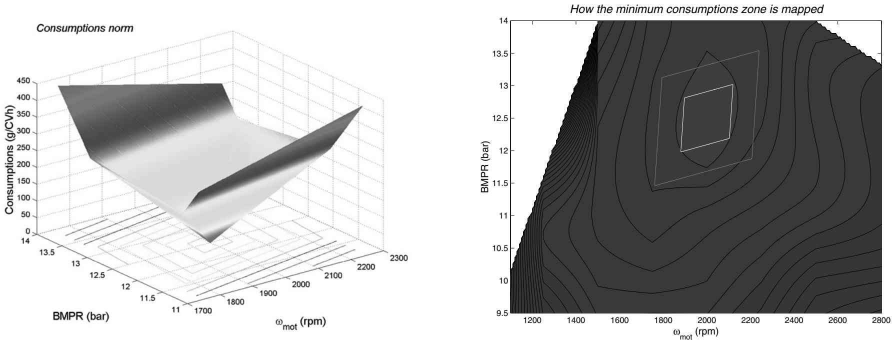


Fig. 7. Piecewise affine approximation of the consumptions map

linear cost function subject to linear constraints, where some of the variables are constrained to be binary, see [8] for details.

5 Simulations

The receding horizon optimal controller based on the hybrid MLD model (2) is simulated in closed loop with a more accurate nonlinear model provided by CRF. The reported simulations are performed using standard speed patterns for emission test cycles, namely the ECE and EUDC patterns[5].

- The ECE cycle is an urban driving cycle, also known as UDC. It was devised to represent city driving conditions, e.g., in Paris or Rome. It is characterized by low vehicle speed, low engine load, and low exhaust gas temperature.
- The EUDC (Extra Urban Driving Cycle) segment has been added after the fourth ECE cycle to account for more aggressive, high speed driving modes. The maximum speed of the EUDC cycle is 120 km/h.

We investigate the behavior of the nonlinear model provided by CRF in closed-loop with three different types of controllers: (1) receding horizon hybrid optimal controller, (2) controller based on static maps (provided by CRF), and (3) gear shifting sequence provided by the EU standard.

The first controller, as described in Section 4, has two degrees of freedom: the gear and the deviation from the nominal requested engine torque. By varying the weights ρ_c and ρ_e it is possible to emphasize the reduction of consumptions or emissions, or in general to trade off between them. The second one is based on a static map provided by CRF that is mainly designed for minimizing consumptions (other maps may be available for minimizing emissions). The third simulation is obtained by feeding the gear shifting sequence provided by the EU standard to the nonlinear vehicle model. Such a sequence represents an ideal

[5] The cycles definition can be found in the EEC Directive 90/C81/01.

sequence, specific for the emission test cycles at hand, and has the objective of reducing both consumptions and emissions.

We underline that only our controller is allowed to modify the nominal engine torque. Unfortunately, since varying the engine torque implies to vary also the speed and since in ECE and EUDC cycles the speed tracking is an important aspect, the deviation Δ_{Torque} from the desired nominal engine torque is highly penalized. As we did not model the vehicle "start up" phase, in all simulation tests rather than decreasing the speed up to 0 km/h it is decreased up to 8 km/h.

Besides the weights, the other two main parameters of the supervisor to be tuned are:

Horizon length T. By increasing the prediction horizon T the controller performance improves, but, at the same time, the number of constraints in (3) (and the complexity of the piecewise affine controller) increases. Therefore, tuning T amounts to find the smallest value which leads to a satisfactory closed-loop behavior. In our case, since the requested engine torque is immediately applied (see Section 2), the engine torque dynamics is neglected, so that the difference in performance using different horizons T (we tested $T = 1, \ldots, 4$) is minimal. Hence, for the benefit of computational simplicity, we chose $T = 1$.

Control signal Δ_{Torque}. While this should be as much as possible close to zero for the reasons mentioned above, it improves the performance of the MPC controller, as it gives the possibility of further reducing consumptions and emissions, at the price of a loss of perfect power tracking, as shown in Fig. 8.

In Fig. 8 and in Fig. 9 we show the simulation results on the ECE and EUDC cycles. In simulating the MPC controller, rather than looking for a trade off between consumptions and emissions, we emphasize the performance where the goal is only to reduce consumptions or only to reduce emissions, as requested by CRF for a comparison between the MPC controller and the one based on static maps.

The MPC controller has a good performance in both cases: clearly, in Fig. 9 the results on the left side are obtained using a high ratio ρ_c/ρ_e (controller MPC$_C$), on the contrary, with a low ρ_c/ρ_e we obtained the results shown on the right side (controller MPC$_E$), by consequently reducing only emissions. By properly choosing the weights we would have a behavior very similar to the one that used the ideal gear shifting sequence.

As expected, when we emphasize the performance where the goal is only to reduce consumptions (MPC$_C$) or emissions (MPC$_E$), the other variable (emissions or consumptions, respectively) sensibly increases, as shown in Fig.10. We remark again that in the present MPC setup one directly selects the desired tradeoff between consumptions and emissions by simply choosing the ratio ρ_c/ρ_e.

The results discussed above were simulated in about 220 s (ECE cycle) and 440 s (EUDC cycle) on a PC Pentium III 1 GHz running MATLAB/Simulink and the MILP solver of Cplex [16], using a prediction horizon $T = 1$ (see [17] for more simulation results). Therefore, the controller is not directly suitable for

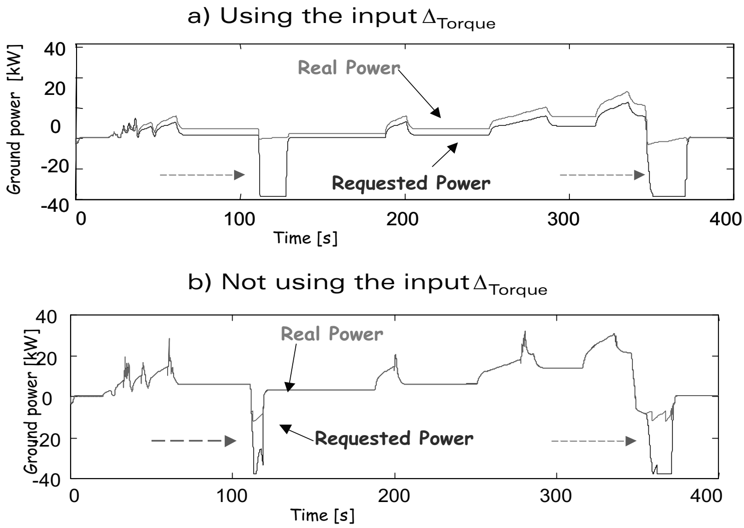


Fig. 8. Comparison between the track traction power with or without using the control signal Δ_{torque}. Dashed arrows show the points, during both simulations, where it was impossible for the engine to reach the desiderate ground power.

implementation on automotive hardware, both for excessive CPU requirements and software complexity. This problem is dealt with in next section.

6 Implementation as a Piecewise Affine Control Law

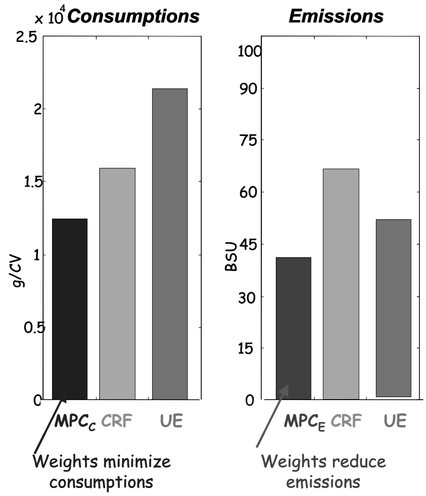
Once the tuning of the MPC controller is done in simulation, the explicit piecewise affine form of the control law can be computed off-line by using a multi-parametric mixed integer linear programming (mp-MILP) solver, according to the approach of [8], [18]. Rather than solving the MILP (3) *on line* for the given current states and reference signals, the idea is to use the mp-MILP solver to compute *off line* the solution of the MILP (3) for all the states and reference signals within an (overestimate of the) expected range of values.

As shown in [8], the control law has the piecewise affine form

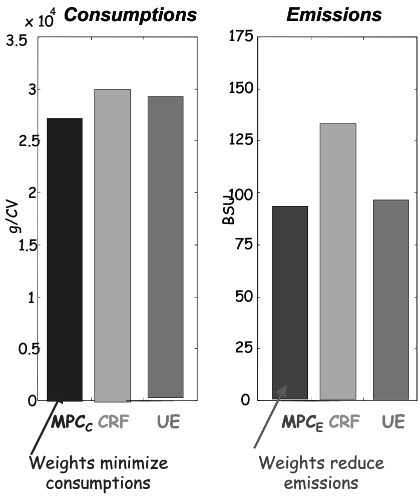
$$u(t) = F_i\Theta(t) + g_i \;\text{ if } H_i\Theta(t) \leq k_i,\; i = 1, ..., n_r, \tag{4}$$

where for our model $u = [\Delta_{torque}, gear^6]'$ and the set of parameters $\Theta = [speed, \%gas\ pedal]'^7$. Therefore, the set of states+references is partitioned into n_r polyhedral cells, and an affine control law is define in each one of them.

[6] It is a real variable; it easy to translate it in the form presented in Section 3.
[7] These parameters can be translated into the state vector by using suitable transformation maps.



(a) Results on the ECE cycle



(b) Results on the EUDC cycle

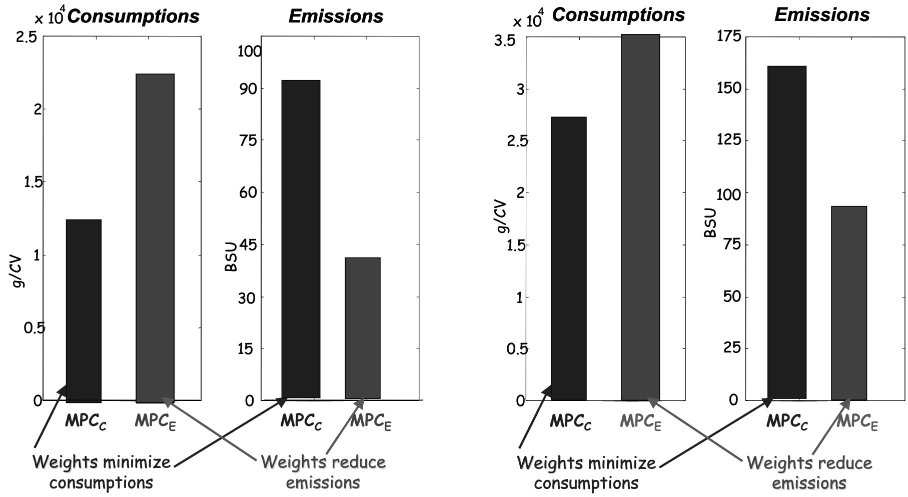
Fig. 9. Comparison of resulting consumptions and emissions using the three different strategies (ECE and EUDC cycles)

We remark that for any given $\Theta(t)$ the on-line solution of RHC via MILP and the explicit off-line solution (4) provide the same result. Therefore, a good design strategy consist of tuning the MPC controller using simulation and on-line optimitazion, and then to convert the controller to its piecewise affine explicit form. The explicit controller will behave in exactly the same way at a much lower computational cost. The control law can in fact be implemented on-line in the following simple way:

i. determine the i-th region that contains the current vector $\Theta(t)$;
ii. compute the $u(t) = F_i\Theta(t) + g_i$ according to the corresponding i-th control law.

More efficient ways of evaluating piecewise affine control law, based on the organization of the controller gains an a balanced search tree, are reported in [19]. At this stage the complexity of the explicit piecewise affine control low (4) has not been yet analyzed. This will be the subject of future research.

Acknowledgements. We thank Domenico Mignone, Fabio D. Torrisi, Nicolò Giorgetti, and Francesco Borrelli for their help with the simulation setup, and A.T.A. (Automotive Technical Association) and the Automatic Control Laboratory at ETH Zurich for the support of Massimo Mannelli.



(a) Comparison on the ECE cycle

(b) Comparison on the EUDC cycle

Fig. 10. Comparison of resulting consumptions and emissions in the MPC controller using different weights (ECE and EUDC cycles)

References

1. Borodani, P., Gianoglio, R., Giuliano, F., Ippolito, L., Lupo, M.: The interactive robotized gearbox improves fuel economy and driver expectations. In: Proc. Fédération Internationale des Sociétés d'Ingénieurs des Techniques de l'Automobile (FISITA), Paris (1998)
2. Ippolito, L., Rovera, G.: Potential of robotized gearbox to improve fuel economy, Certosa di Pontignano (SI), Italy (1996)
3. Hoio, Y., Iwatsuki, K.: Toyota five speed automatic transmission with application of modern control theory. SAE Paper 920610 (1992)
4. Naruse, T., Nakashima, Y.: A study on evaluation method and improvement of shift quality of automatic transmission. SAE Paper 930673 (1993)
5. Yamaguchi, H., Takahashi, H.: Automatic transmission gear change control using fuzzy logic. (Nissan Motor Company Ldt.)
6. Borodani, P., Gianoglio, R., Giuliano, F., Lupo, M.: Gear change control system according to driving style in a motor vehicle provided with servo assisted gears. European patent nr. EP 0950839 (1999) International Application Number: 99107256.2 3.
7. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica **35** (1999) 407–427
8. Bemporad, A., Borrelli, F., Morari, M.: Piecewise linear optimal controllers for hybrid systems. In: Proc. American Contr. Conf., Chicago, IL (2000) 1190–1194
9. Genta, G.: Motor Vehicle Dynamics: Modeling and Simulation. Volume 43 of Series on Advances in Mathematics for Applied Sciences. World Scientific Pub Co (1997)

10. Gillespie, T.: Fundamental of Vehicle Dynamics. Society of Automotive Engineers, Inc. (1992)
11. Antsaklis, P.: A brief introduction to the theory and applications of hybrid systems. Proc. IEEE, Special Issue on Hybrid Systems: Theory and Applications **88** (2000) 879–886
12. Lygeros, J., Tomlin, C., Sastry, S.: Controllers for reachability specifications for hybrid systems. Automatica **35** (1999) 349–370
13. Branicky, M.: Studies in hybrid systems: modeling, analysis, and control. PhD thesis, LIDS-TH 2304, Massachusetts Institute of Technology, Cambridge, MA (1995)
14. Balluchi, A., Benvenuti, L., Benedetto, M.D., Pinello, C., Sangiovanni-Vincentelli, A.: Automotive engine control and hybrid systems: Challenges and opportunities. Proc. IEEE **88** (2000) 888–912
15. Torrisi, F., Bemporad, A.: HYSDEL — A tool for generating computational hybrid models. Technical Report AUT02-03, ETH Zurich, Submitted for publication on IEEE Transactions on Control Systems Technology (2002) `http://control.ethz.ch/~hybrid/hysdel`.
16. ILOG, Inc.: CPLEX 7.0 User Manual, Gentilly Cedex, France. (2000)
17. Mannelli, M.: An automotive gear-shift decision system: modeling and hybrid control (2002) Master thesis. University of Florence, Italy. In Italian.
18. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.: The explicit linear quadratic regulator for constrained systems. Automatica **38** (2002) 3–20
19. Tøndel, P., Johansen, T., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. Automatica (2003) In press.

A Appendix

Below we report the HYSDEL model of the vehicle, from which we obtain the MLD model. Note that T_{ref}, T_{brake}, *slope* are treated here as measured constant states, as their value is updated at every step.

```
/* Model 8: model for the control release 8.5
27.02.02(M.S.G)-01.03.02(N.S.G.) M.p.T */

SYSTEM MODEL8 {
 INTERFACE
 {
  STATE
  {
    REAL wsec, Cref, brake, slope;
    /* wsec  =  The only state in the reduced-order model: it has not a physical meaning
       Cref  =  Engine reference torque at time t
       brake =  Braking torque
       slope =  Slope*/
  }
  INPUT
  {
    REAL DC;
    /* DC= Engine torque variation*/
    BOOL gear1,gear2,gear3,gear4;
    /*gear(i)= i-th gear; gear 5 is obtained when each input is zero*/
  }
  OUTPUT
  {
    REAL wmot, torque;
    /* wmot  = Primary shaft speed in rpm*
       torque= Real torque applied*/
  }
  PARAMETER
  {
        REAL T1 = 3.909;
        REAL T2 = 2.238;
        REAL T3 = 1.444;    /*Gear ratio (for each gear engaged)*/
        REAL T4 = 1.029;
```

```
      REAL T5 = 0.767;
      REAL Tp = 3.15;     /*Bridge ratio*/
      REAL wmin=700;      /*Minimum primary shaft speed*/
      REAL wmax=6000;     /*Maximum primary shaft speed*/

      REAL C1=0.8874;
      REAL C2=1.0026;
      REAL C3=1.0442;    /*coefficients for the outputs re-establish from the state (of the
                           reduced-order model) wsec*/
      REAL C4=1.0601;
      REAL C5=1.0676;
      .
      .                  /*other parameters are omitted for lack of space*/
      .
      REAL pi= 3.14159;
      REAL e = 1e-6;     /*precision*/
 }
}
IMPLEMENTATION
{
  AUX
 {
   REAL  wsec1, wsec2, wsec3, wsec4, wsec5, Cm1, Cm2, Cm3, Cm4, Cm5, Cwmax1, Cwmax3;
   BOOL w1, w2;     /*auxiliary variables*/
 }
 AD
 {
   w1 = wmax1-(60/2/pi)*(C1*wsec1*T1+C2*wsec2*T2+C3*wsec3*T3+C4*wsec4*T4+C5*wsec5*T5)<=0
                                  [wmax1-wmin,wmax1-wmax,e];
   w2 = wmax2-(60/2/pi)*(C1*wsec1*T1+C2*wsec2*T2+C3*wsec3*T3+C4*wsec4*T4+C5*wsec5*T5)<=0
                                  [wmax2-wmin,wmax2-wmax,e];
   /*w3 = wmax3-wsec1*T1+wsec2*T2+wsec3*T3+wsec4*T4+wsec5*T5<=0 [wmax3-wmin,wmax3-wmax,e];*/
 }
 DA
 {
   wsec1 = {IF (gear1)             THEN a111*wsec+b111*rend*(T1*Cref+DC)+b112*slope+b113*brake+b114*Crot+b115
                           [(wmax*pi*2)/(60*T5),0,e]};
   wsec2 = {IF (gear2)             THEN a211*wsec+b211*rend*(T2*Cref+DC)+b212*slope+b213*brake+b214*Crot+b215
                           [(wmax*pi*2)/(60*T5),0,e]};
   wsec3 = {IF (gear3)             THEN a311*wsec+b311*rend*(T3*Cref+DC)+b312*slope+b313*brake+b314*Crot+b315
                           [(wmax*pi*2)/(60*T5),0,e]};
   wsec4 = {IF (gear4)             THEN a411*wsec+b411*rend*(T4*Cref+DC)+b412*slope+b413*brake+b414*Crot+b415
                           [(wmax*pi*2)/(60*T5),0,e]};
   wsec5 = {IF ~(gear1|gear2|gear3|gear4)  THEN
   a511*wsec+b511*rend*(T5*Cref+DC)+b512*slope+b513*brake+b514*Crot+b515 [(wmax*pi*2)/(60*T5),0,e]};

   Cm1 = {IF (gear1)           THEN Cref+DC/T1        [Cmax,-70,e]};
   Cm2 = {IF (gear2)           THEN Cref+DC/T2        [Cmax,-70,e]};
   Cm3 = {IF (gear3)           THEN Cref+DC/T3        [Cmax,-70,e]};
   Cm4 = {IF (gear4)           THEN Cref+DC/T4        [Cmax,-70,e]};
   Cm5 = {IF ~(gear1|gear2|gear3|gear4)    THEN Cref+DC/T5        [Cmax,-70,e]};

   Cwmax1 = {IF (~w1) THEN   /* Maximum Engine torque in the range [700, w1] rpm*/
   Tcmax1*((C1*wsec1*T1+C2*wsec2*T2+C3*wsec3*T3+C4*wsec4*T4+C5*wsec5*T5)-(wmax1)*(2*pi/60))+Cmax
                           [9000,-5,e]};

   Cwmax3 = {IF w2 THEN      /* Maximum Engine torque in the range [w2, 6000] rpm*/
   Tcmax3*((C1*wsec1*T1+C2*wsec2*T2+C3*wsec3*T3+C4*wsec4*T4+C5*wsec5*T5)-(wmax2)*(2*pi/60))+Cmax
                           [500,-50,e]};
 }

 CONTINUOUS
 {

   wsec=wsec1+wsec2+wsec3+wsec4+wsec5; /*in rad/s*/
   Cref=Cref;                          /*Nm*/
   brake=brake;                        /*Nm*/
   slope=slope;                        /*N*/
 }

 OUTPUT
 {
   wmot=(C1*wsec1*T1+C2*wsec2*T2+C3*wsec3*T3+C4*wsec4*T4+C5*wsec5*T5)*60/2/pi;
   /*Primary shaft speed in rpm*/
   torque=4*pi*(Cm1+Cm2+Cm3+Cm4+Cm5)/(100*1.91);
   /*Engine Torque/swept volume in BMRP*/
 }
 MUST
 {
   -brake<=0;         /* Minimum brake torque*/
   brake<=maxbrake;   /* Maximum brake torque*/
   /*~((~w1)&(~(gear1|gear2|gear3|gear4)));*/
   w2->w1;
   /*w3->w2;
   w3->w1*/

   -((REAL gear1)+(REAL gear2)+(REAL gear3)+(REAL gear4)+1)<=-0.9999;
   (REAL gear1)+(REAL gear2)+(REAL gear3)+(REAL gear4)<=1.0001;

   /* Check the Primary shaft speed*/
```

```
    -(C1*wsec1*60*T1+C2*wsec2*60*T2+C3*wsec3*60*T3+C4*wsec4*60*T4+C5*wsec5*60*T5)/(2*pi)<=-wmin;
    C1*wsec1<=((wmax-1000)*pi*2)/(60*T1);
    C2*wsec2<=(wmax*pi*2)/(60*T2);
    C3*wsec3<=(wmax*pi*2)/(60*T3);
    C4*wsec4<=(wmax*pi*2)/(60*T4);
    C5*wsec5<=(wmax*pi*2)/(60*T5);

    /*Maximum Engine Torque*/
    Cm1<=Cwmax1+Cmax*((REAL w1)+(1-(REAL w2))-1)+Cwmax3;
    Cm2<=Cwmax1+Cmax*((REAL w1)+(1-(REAL w2))-1)+Cwmax3;
    Cm3<=Cwmax1+Cmax*((REAL w1)+(1-(REAL w2))-1)+Cwmax3;
    Cm4<=Cwmax1+Cmax*((REAL w1)+(1-(REAL w2))-1)+Cwmax3;
    Cm5<=Cwmax1+Cmax*((REAL w1)+(1-(REAL w2))-1)+Cwmax3;

    /*Constraints for Delta Torque*/
     DC<=DCmax;
    -DC<=-DCmin;
  }
 }
}
```