Delft University of Technology

Delft Center for Systems and Control

Technical report 07-022

Comparison of a linear and a hybrid adaptive cruise controller for a SMART[*]

D. Corona and B. De Schutter

If you want to cite this report, please use the following reference instead:

D. Corona and B. De Schutter, "Comparison of a linear and a hybrid adaptive cruise controller for a SMART," *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, Louisiana, pp. 4779–4784, Dec. 2007.

---

[*]This report can also be downloaded via https://pub.bartdeschutter.org/abs/07_022.html

Comparison of a Linear and a Hybrid Adaptive Cruise Controller for a SMART

Daniele Corona and Bart De Schutter

Abstract— An adaptive cruise controller (ACC) is a device used in modern automotive applications that aims to achieve the tracking of a leading vehicle, allowing safety, comfort driving and overall improvement of traffic streams. Through appropriate modeling and design it permits also to address other tasks such as energy saving, environmental protection and reduction of the mechanical stress. For a type of small car, a SMART, we propose the design of an ACC using control methods based on tuned proportional-integral (PI) action and on model predictive control (MPC), developed for linear and piecewise affine (PWA) systems. We implement these controllers and compare the main properties and strong/weak points of each method, collecting the comparison keys in a table. The trade-off between complexity and accuracy of the solutions is also discussed.

I. Introduction

An adaptive cruise controller (ACC) is a modern device embedded in several vehicles branches that aim to increase the road safety and passenger comfort. Some authors [1] claim that it can aid in maximizing the throughput of a road section, thus improving the traffic flow. It can also contribute to less constraining, yet not less relevant, issues such as fuel consumption (related to economy and pollution) and reduction of the long-term mechanical stress. A natural way to meet these expected behaviors is to chose a control strategy based on constrained optimization processes like MPC schemes [2].

The purpose of this paper is to present some of the results we have obtained within a wide comparison study [3] of possible control methods for an ACC of a 6 gears, 37 kW gasoline SMART. This study considers the ACC design as a benchmark problem for existing MPC-PWA methods. The engine torque, the air drag nonlinearities and the event-driven gearbox, as well as the presence of design constraints, make the task suitable for the MPC framework. In this paper we compare three different representative control schemes. The first exploits the hybrid behavior of the gearbox and a PWA approximation [4] of the air drag, leading to a control problem with mixed integer control variables [5]. We propose a PWA representation of the system and of its corresponding MLD (mixed logical dynamics [5]) model, which allows to transform the MPC into a sequence of *online* MILPs (mixed integer linear programs).

The second method is MPC-based as well, but it uses a less accurate prediction model, aiming to trade off the accuracy of the solution for a faster controller. In particular

D. Corona and B. De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. B. De Schutter is also with the Marine and Transport Technology department, Delft University of Technology. d.corona@tudelft.nl,b@deschutter.info

it approximates the nonlinear air drag force with affine gain scheduled modes and the engine traction force is taken to be a constant averaged value over all gears. Moreover we give the explicit solution over the design parameters and we calculate offline a polyhedral partition as in [6].

The third method designs the ACC as it is currently done in industry [7]. It is a PI (proportional integral) based approach, where the coefficients are Gaussian-like functions of the current tracking error [7]. The three parameters of the Gaussian functions are computed offline minimizing the same performance index as for the first two methods under a standard nominal scenario.

The paper is organized as follows: we first describe, in Section II the simulation model, involving the engine and friction nonlinearities and the event-driven gearbox, the specific constraints and the control goals. Then, in Section III, we provide a short description of three different control methods. The target is to highlight the major advantages/disadvantages of each control law. To this purpose we perform, in Section IV, numerical simulations under a specific nominal scenario and establish a comparison table that collects the results on some key aspects of the control design, of the complexity and of the quality of the solution.

II. Model and problem description

A. Set-up

The ACC controls the traction/brake dynamics, aiming to keep a minimal separation and a speed adaptation with a preceding vehicle. Two relevant realizations of the device can be distinguished. One is for the platooning [7], [8], with large use in freight road transportation, and the other is a stand-alone version, mainly oriented as a private usage of the vehicle. In the first case a communication protocol is established among the vehicles and the local controllers are aware of the operating conditions of the others. In the second case the vehicle is autonomous and, based on measurements taken or received from the environment, it will take its own decision. This study is focused on the first scenario, where one car (a SMART) is aiming to track another vehicle, as in Figure 1.

B. Model

In the case we consider (see also Figure 1), it is allowed to restrict the attention to the rear vehicle, modeled by the following differential equation:

$$m\ddot{s}(t) + (c\dot{s}^2(t) + \mu\,mg)\mathrm{sgn}(\dot{s}(t)) = b(j,\dot{s})u(t) - mg\sin(\xi(s)), \quad (1)$$
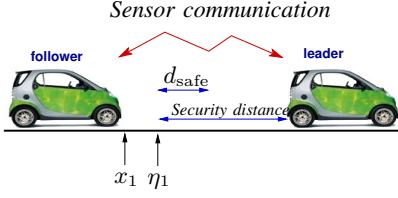
Fig. 1.   ACC set-up.

TABLE I

Definitions and values of the entries of (1).

Parameter	Description	Value
m	Mass of vehicle	$800\ kg$
R	Average wheel radius	$0.28\ m$
c	Viscous coefficient	$0.5\ kg/m$
μ	Coulomb friction coeff. (dry asphalt)	0.01
g	Gravity acceleration	$9.8\ m/s^2$
ξ	Slope of the road	$0\ rad$
$w_{\min}$	Minimum engine rotational speed	$105\ rad/s$
$w_{\max}$	Maximum engine rotational speed	$630\ rad/s$

TABLE II

Values of the transmission rate, the maximum traction force and the switching condition per gear.

Gear j	$p(j)$	$b(j)\ (N)$	$v_L\ (m/s)$	$v_H\ (m/s)$
I	14.203	4057	0	9.46
II	10.310	2945	5.43	13.04
III	7.407	2116	7.56	18.15
IV	5.625	1607	9.96	23.90
V	4.083	1166	13.70	32.93
VI	2.933	838	19.10	45.84

TABLE III

Values of the parameters specifying the constraints.

Parameter	Description	Numerical value	Unit
$x_{1,\min}$	Minimum position	0	m
$x_{1,\max}$	Maximum position	3000	m
$x_{2,\min}$	Minimum velocity	2.0	m/s
$x_{2,\max}$	Maximum velocity	40.0	m/s
d_{safe}	Security pos. overshoot	10.0	m
a_{acc}	Comfort acceleration	2.5	m/s^2
a_{dec}	Comfort deceleration	2.0	m/s^2
$u_{\max}$	Maximum throttle/brake	1	—

where $s(t)$ is the position of the rear vehicle at time t and $b(j,\dot{s})$ is the traction force coefficient of the normalized throttle/brake input $u(t)$, ξ is the slope of the road; the terms on the left-hand side indicate the inertia, the air drag and the road-tire Coulomb friction. The numerical values of m, c, μ, ξ and g are listed in Table I. The value of the function $\text{sgn}(\cdot)$ is equal to 1, 0 or -1 when its argument is positive, zero or negative respectively. The traction force coefficient $b(j,\dot{s})$ is the engine torque curve, in the rotational velocity range $w \in [w_{\min}, w_{\max}]$, transmitted through the gearbox and acting in the contact area between road and tires. More specifically we have $b(j,\dot{s}) = \frac{T_e(w)p(j)}{R}$ with $\dot{s} = \frac{wR}{p(j)}$, where $T_e(w)$ is the engine torque, R is the average radius of the wheels, $p(j)$s are the gear ratios, given in Table II. From [9] the engine torque is constant ($T_{e,\max} = 80\ Nm$) for $w \in [200, 480]\ rad/s$. In Table II we give the values of $p(j)$ and $b(j)$ in this specific range. The function $b(j,\dot{s})$ in the full range of velocity is depicted in Figure 2.

Braking will be simulated by applying a negative throttle. Due to friction behavior in motion inversion [10], model (1) is valid as long as the ground speed $\dot{s}$ is different from
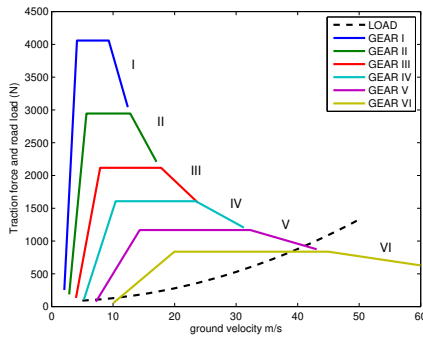

Fig. 2.   *Traction force per gear transmitted to the wheel at the maximum throttle.*

zero, hence we impose $\dot{s}$ to be above a nonzero minimum velocity (3).

A state space representation of system (1) is:

$$\dot{x} = f(x) + B(j,x)u, \qquad (2)$$

where $x \triangleq [s, \dot{s}]^{\mathrm{T}} = [x_1, x_2]^{\mathrm{T}}$, and

$$f(x) = \begin{bmatrix} x_2 \\ -\frac{c}{m}x_2^2 - \mu g \end{bmatrix} \qquad B(j,x) = \begin{bmatrix} 0 \\ -\frac{b(j,x_2)}{m} \end{bmatrix}.$$

This model is nonlinear because of the quadratic term of the friction and of the traction force, and hybrid, because of the gears.

C. Constraints

Safety, comfort and economy or environmental issues, as well as limitations on the model, result in defining operating constraints. We consider constraints on the state x, on the control input u, and on the gear j. Numerical values for the parameters appearing in these constraints are listed in Table III.

(i) State constraints: given $\eta \triangleq [\eta_1, \eta_2]^{\mathrm{T}}$ the reference position and velocity of the reference, we should have,

$$\begin{aligned} x_{2,\min} &\leq x_2(t) \leq x_{2,\max} \\ x_1(t) &\leq \eta_1(t) + d_{\text{safe}} \qquad \forall t \geq 0 \\ a_{\text{dec}} &\leq \dot{x}_2(t) \leq a_{\text{acc}}. \end{aligned} \qquad (3)$$

The above equations express, respectively, the operational range of the speed, the collision avoidance with a given tolerance d_{safe} (Figure 1), and bounds on the acceleration. Without loss of generality we shall consider an additional *non-operational* constraint on the position, $x_{1,\min} \leq x_1(t) \leq x_{1,\max}$, for all $t \geq 0$, which is necessary to implement the first control method.

(ii) Input constraints: $|u(t)| \leq u_{\max}$, for all $t \geq 0$.

(iii) Gear shift constraints: given $j \in \{1, \ldots, 6\}$ the gear shift position should satisfy, for all $t \geq 0$, $1 \leq j(t) \leq 6$ and $|j(t + dt) - j(t)| \leq 1$, where dt is a finite small time interval. The latter forbids jumps of gears, which would provoke non-optimal fuel consumption in up-shifting and mechanical stress in down-shifting.

A bounded range of velocities is valid per each gear, hence,

$$v_L(j) \leq x_2(t) \leq v_H(j), \quad \forall t \geq 0. \tag{4}$$

where the values of $v_L(j), v_H(j)$ are given in Table II.

For the sake of simplicity we propose a linearized approximation of (4)

$$v_0 + v_1 j \leq x_2(t) \leq v_0 + v_1(j + 1), \quad \forall t \geq 0, \tag{5}$$

where the values of v_0 and v_1 are obtained as

$$(v_0, v_1) \triangleq \arg \min_{v_0, v_1} \sum_{j=1}^{6} \Big(\gamma_L \big(v_L(j) - (v_0 + jv_1) \big)^2 + \gamma_H \big(v_H(j) - (v_0 + (j+1)v_1) \big)^2 \Big),$$

subject to $v_0 + v_1 \geq x_{2,\min}$. The choice of the weights γ_L, γ_H was preferred towards the higher velocities ($\gamma_L = 1, \gamma_H = 100$), where the engine works with higher efficiency.

Although some of these constraints may be violated without causing major damages, i.e., collision or engine breakdown, we decided to consider all of them as *hard*.

D. Control problem

A more natural way of considering the problem deals with the discrete time sampling. In this application, in fact, the measurements and the data flow from the front vehicle take place in a given time period T. For this reason we will consider, from now on, a discrete time version of the problem with time variable k. In this set-up the measured output $x(k)$, possibly affected by disturbances $\Omega(k)$, is plugged into the controller, which also receives the prediction of the reference $\eta(k)$. According to these values, the controller computes the next control input that is fed into the simulation model.

The main goal is to design a control signal $u(k)$ and a gear shift sequence $j(k)$ that trade off the reduction of the tracking error $\varepsilon(k) \triangleq x(k) - \eta(k)$ for the variation of the continuous input $\Delta u(k) \triangleq u(k) - u(k-1)$ and the number of gears switchings $\Delta j(k) \triangleq j(k) - j(k-1)$, within the constraints described above.

III. DESIGN METHODS

Three different representative controllers are selected to perform the comparison: two of them are MPC based (an online and an offline one) with different prediction models, the third one is adopted from the industrial realization of ACC devices and it is a specifically tuned PI:

- Online PWA MPC: MLD-on (Section III-A);
- Basic gain scheduling approximation: BGS (Section III-B);
- Optimized proportional-integral (PI) controller (Section III-C);

In the first method we implement an MPC controller using as a prediction model a PWA approximation of the system. This permits to transform the problem into an online mixed integer linear program based on the equivalent MLD formulation. The MPC approach is largely used to design the control actions of constrained systems and in particular PWA systems (see e.g. [5], [11], [12]). In the second case we use a very simple prediction model, obtained via a gain scheduling of the nonlinear air drag. The gear shift action is not considered and the traction force B_j is averaged over j. For this method we propose the parametric offline solution. The third method tunes a PI-based controller whose parameters depends on the position and velocity errors. The PI coefficients are determined offline in order to meet a specific optimization criterion.

For the first two methods we provide the expression of a norm 1 MPC performance index. The control signal $u(k)$ is designed by solving a receding horizon constrained finite time optimal control problem. The prediction or acquisition of N_p samples ahead of the front vehicle trajectory is used to compute the optimal control law $u(k)$ by solving

$$\min_{\tilde{u}(N_p), \tilde{j}(N_p)} J(\theta(k), \tilde{u}(N_p), \tilde{j}(N_p)) \triangleq \sum_{i=1}^{N_p} ||Q_x \varepsilon(k+i)||_1 + ||Q_{\Delta u} \Delta u(k+i-1)||_1 + ||Q_{\Delta j} \Delta j(k+i-1)||_1, \tag{6}$$

subject to a prediction model and the given constraints. Here $\tilde{u}(N_p) \triangleq [u(k), \ldots, u(k + N_p - 1)]^T$ is the sequence of control inputs, $\tilde{j}(N_p) \triangleq [j(k), \ldots, j(k + N_p - 1)]^T$ the gear shift sequence, $Q_x, Q_{\Delta u}$ and $Q_{\Delta j}$ are weight matrices, $\theta(k) \triangleq [x(k)^T, u(k-1), j(k-1), \eta(k+1)^T, \ldots, \eta(k + N_p)^T]^T$ is a set of parameters. A shorter control horizon $N_c < N_p$ may also be considered, i.e., $\Delta u(k+i) = 0$, $i = N_c, \ldots, N_p - 1$, $\Delta j(k+i) = 0$, $i = N_c, \ldots, N_p - 1$. This choice has the general advantage of reducing the number of variables and of providing a smoother solution. Nevertheless here we only consider $N_p = N_c$ as it is out of the scope of this paper to investigate the behavior of the same controller for different tunings.

A. Method 1: Piecewise affine MPC (MLD-on)

In order to give the PWA representation of the model described in Section II-B we approximate the function $b(j, x_2)$, the second component of the vector $B(j, x)$ in (2), as follows. We assume it is linear dependent from the gear j and independent from the velocity x_2. This last assumption is acceptable in a wide velocity range as shown in Figure 2. Hence

$$b(j, x_2) \approx b(j) \approx b_j \triangleq \beta_0 + j\beta_1 \tag{7}$$

where $j \in \{1, \ldots, 6\}$ is the gear, $\{b(1), \ldots, b(6)\}$ are the maxima in Figure 2, listed in Table II, and

$$(\beta_0, \beta_1) \triangleq \arg \min_{\beta_0, \beta_1} \sum_{j=1}^{6} \big(b(j) - (\beta_0 + j\beta_1) \big)^2.$$

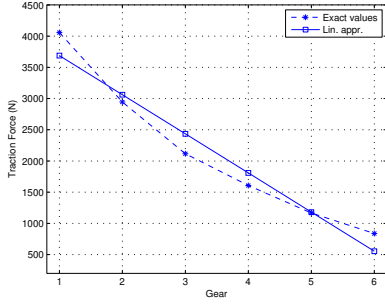This approximation is shown in Figure 3.

Fig. 3. *Traction force approximation for different gears.*

TABLE IV
*Encoding of the value of the gear j via three binary variables δ_i,
with $i = 1, 2, 3$.*

Gear j	δ_1	δ_2	δ_3
I	0	0	0
II	1	0	0
III	0	1	0
IV	1	1	0
V	0	0	1
VI	1	0	1

The integer value of the gear can be encoded using three binary variables, $\delta_i \in \{0, 1\}$, with $i = 1, 2, 3$, as listed in Table IV.

With this abstraction the traction force (7) becomes

$$b_j = (\beta_0 + \beta_1) + \beta_1 \delta_1 + 2\beta_1 \delta_2 + 4\beta_1 \delta_3, \quad (8)$$

which integrates continuous (u) and logic variables (δ_i) as in [5].

With similar arguments we approximate the function $-\frac{c}{m} x_2^2$, the second component of the vector $f(x)$ in (2), with two affine modes, as depicted in Figure 4. The PWA representation of the system is therefore

$$x(k+1) = \begin{cases} A_1 x(k) + F_1 + B_j u(k), & \text{if } x_2(k) < \alpha \\ A_2 x(k) + F_2 + B_j u(k), & \text{if } x_2(k) \geq \alpha, \end{cases} \quad (9)$$

where the matrices A_1, A_2, F_1, F_2 are derived using the data shown in Figure 4, and $B_j = [0, b_j]^{\text{T}}$.

To complete the MLD transformation we use an additional binary variable that indicates the mode in 9. After some manipulations, similar to the ones in [5], the equivalent MLD
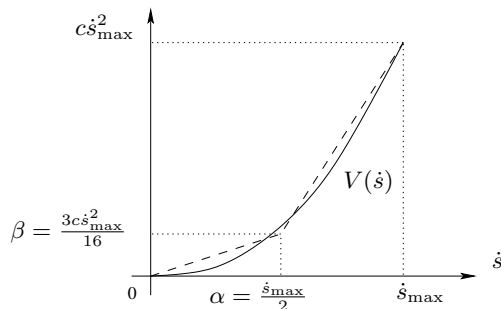

Fig. 4. *Nonlinear friction (solid), PWA approximation (dashed).*

model takes the form of

$$x(k+1) = Ax(k) + Bv(k) + F, \quad (10)$$

where the vector $v(k)$ has binary and continuous components. Plugging this model on the MPC norm 1 performance index and in the constraints we obtain a MILP problem of the form

$$J(\theta(k))^* = \begin{aligned} &\min_{\tilde{y}} && c^{\text{T}}\tilde{y} \\ &\text{s.t.} && E\tilde{y} \leq G + E_\theta \theta(k), \end{aligned} \quad (11)$$

where $\tilde{y}$ contains the *prediction* of the control variable $\tilde{v}$ and some additional auxiliary variables. The matrices E, G and E_θ constrain the search space of the optimization problem in order to meet the physical and the logic constraints introduced by the MLD transformation.

B. Method 2: Basic gain scheduling approximation (BGS)

In this method the nonlinear curve depicted in Figure 4, is approximated into 6 linear models m_i, $i = 1, \ldots, 6$ in point to point secant approximation[1]. Additionally the function $b(j, x_2)$ is approximated with its average values over all gears, thus the problem becomes linear as better detailed in [3].

For each affine model m_i we solve an offline mpLP [6], [13] problem with a similar structure as (11), but with differently defined entries $\tilde{y}, c, E, G, E_\theta$; in particular, the variable $\tilde{y}$ is purely continuous. The offline solution of this problem constructs a polyhedral partition of the parameter space. Each region $\mathcal{R}_i$ of the partition stores an affine controller $u_i(k) = K_i \theta(k) + H_i$. During the simulation the region $\mathcal{R}_i$ is selected according to the current value of the parameters $\theta(k)$, and the corresponding controller $u_i(k) = K_i \theta(k) + H_i$ is plugged into the system. The gear is selected using (5).

C. Method 3: Proportional-integral action (PI)

The third method is based on a proportional-integral (PI) action. This is the technique mostly used in practice [7]. The controller first computes a desired acceleration

$$a_{\text{d}}(k) = k_I \varepsilon_1(k) + k_P \varepsilon_2(k), \quad (12)$$

where k_P, k_I are the proportional and integral coefficients and $\varepsilon(k) = x(k) - \eta(k)$ is the tracking error at step k. The actuators regulate the throttle, the gear and the braking action in order to better achieve the desired value of the acceleration.

In industrial versions of the device the coefficients k_P, k_I depend on the current value of the state $x(k)$ (position and velocity) and of the tracking error signal $\varepsilon(k)$, according to specifically designed bell-shaped curves [7]. The parameters of these curves (offset, peak and standard deviation) are tuned empirically to obtain high comfort in acceleration and high security in braking for a variety of scenarios. In this study we tune these parameters so that the *total* cost of the evolution, at a nominal scenario, is minimized. The gear is selected using (5).

---

[1]Without loss of generality, the number of models is chosen equal to the number of gears.

Description	Numerical value
State weight matrix Q_x	$\begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$
Input weight matrix $Q_{\Delta u}$	0.1
Shift weight matrix $Q_{\Delta j}$	0.01
Prediction horizon N_p	2
Control horizon N_c	2
Sampling time T	$1\,s$
Simulation time	$75\,s$
Throttle initial position	0
Initial gear	I
State initial condition	$[0,5]^T$
Leading vehicle speed	$15\,m/s$

IV. NUMERICAL COMPARISON

A. General experimental set-up

We consider a reference trajectory $\eta(t)$ where the front vehicle is driving at the constant velocity. This choice permits to study the behavior of the controllers in a nominal driving scenario, on a horizontal, extra-urban road with low traffic density. The general data common to all experiments are summarized in Table V. We will also consider disturbances. Modern technology (differential GPS, laser sensors and extended Kalman filters [14]) provides fast and highly accurate measurements, up to one meter error in positioning and to $0.1\,m/s$ in velocity. These values will be used to characterize the disturbances.

In these experiments the reference (the leading vehicle) is moving with a constant speed of $15\,m/s$, ($54\,km/h$). The controller measures x, η, and for methods 1 and 2 *receives* the reference in the next $N_p - 1$ future samples. On the basis of the value of the gear and control at the previous time sample it evaluates the new control input and gear, that will be plugged in the simulation model described in Section II-B, integrated with `Matlab ode45`.

The three methods were implemented in `Matlab 7`, `Linux 2.4.22 OS` on an `INTEL Pentium 4, 3GHz` processor. The online MILP is solved with `Cplex` under `TOMLAB v5.1`; the *multi-parametric* problem (BGS) is solved with the `multi-parametric toolbox MPT v2.6` [15] and the optimal choice of coefficients for the PI are obtained via the nonlinear programming subroutine `fmincon` of the `Matlab` optimization toolbox.

B. Comparison issues and results

The comparison issues are listed in Table VI, and for each line of the table the worst entry is indicated in **bold**, while the best case is in *italic*. The comparison is divided into four groups.

1) Computational features: The computational aspects of the problem should give the reader insight into the time and memory demands and complexity of the method. We use the acronym NP-H (P) to indicate Non-Polynomial Hard (Polynomial) complexity. The *maximum* and *average* online times along the whole simulation period are collected. Note

that the linear methods (BGS and PI) are really competitive. We remark here that the sampling time $T = 1\,s$ is longer than in common ACC devices, where measurements are taken at the frequency of 5 to 10 Hz [2], [16] (i.e., $T = 0.1 - 0.2\,s$). Nevertheless this is not restrictive; in fact all methods require an online computation time shorter than $0.1\,s$.

The major advantage of the offline methods is that they *do not* require the optimizer on-board, but merely an efficient data-base browser and enough space to store the solution. In a real-life application this is highly preferable, since the performance of an on-board platform is unquestionably poorer than of a desktop computer. Moreover, the optimizers require extra memory (indicated in Table VI with "+opt.") and may have a license cost impact.

The *Max tractable* N_p, only applicable for MPC methods, is the biggest N_p such that the online computational time is smaller than the sampling time $T = 1\,s$; for the offline MPC this is the biggest N_p such that the required online memory is smaller than $128\,Mb$, the memory capacity of an on-board chip.

The item *Number (#) of regions* of the partition is applicable only for the BGS method and it gives an indication of the complexity of the look-up table.

2) Programming features: This group refers to the *programming features*, such as basic data of the optimization problem, and in particular the size of the problem. The *number of variables* (real and integer), the *number of constraints*, and the number of parameters (i.e., the dimension of $\theta(k)$) are listed. This group shows that the more the accurate the model, the bigger is the size of the problem. In this section of the table we also recall whether the method is online (*Y* = 'Yes', *N* = 'No').

3) Solution features: This group provides a better insight into the physical/mechanical aspects of the solution. The first indicator is the total cost of the evolution in closed loop. A high value of the cost means, broadly speaking, a worst tracking of the reference. For this item the most approximated methods behave better. On the other hand, it can be seen in the following line, they violate the constraint on the acceleration, due to a very aggressive initial action. The PI controller, which is unconstrained, performs the poorest. The same general trend is for the other items. Next we consider transient features: position and velocity overshoot, the duration of the transient on the velocity tracking[2] and the number of gear switchings made to reach the steady state of the velocity. In particular with *position overshoot* we show of how many meters the vehicle overtakes the reference[3].

4) Effect of disturbances: The same conclusion cannot be drawn for the number of constraint violations, in the fourth group of the table: in this case the bigger model mismatch of the linear methods is the source of numerous constraint violations. This shows once more the importance of the trade-off in the MPC framework between the accuracy of the prediction model and the quality of the solution. To better highlight this aspect, the same computations were performed

---

[2]The time required by the controller to keep the velocity within a 5 % band around the reference.

[3]Note that the hard constraint $x_1(k) \le \eta_1(k) + d_{\text{safe}}$ is still satisfied.

TABLE VI

Benchmark problem: comparison issues for the 3 methods.

Method	MLD-on	BGS	PI
Computational features			
Complexity	NP-H	NP-H	NP
Max online time (s)	**0.0521**	0.00048	*0.00019*
Avg. online time (s)	**0.0478**	*0.00007*	0.00013
Offline time (s)	*0.057*	630.52	2.11×10^4
Online mem. (Mb)	**0.46+opt.**	4.09	~ 0
Offline mem. (Mb)	0.09	**0.08+opt.**	~ 0
Max tractable N_{p}	*10*	**5**	–
# regions	–	630	–
Programming features			
Program class	MILP	mpLP	NLP
Online method	Y	N	N
# variables	**32**	*8*	9
# constraints	**102**	*44*	–
# binary vars.	8	–	–
# parameters	–	7	–
Solution features			
Cost of evolution	**120.97**	73.62	104.91
Max acc. (m/s^2)	*2.46*	3.5	**4.93**
Max dec. (m/s^2)	1.79	*1.35*	**2.89**
Max $\Delta u(k)$	*0.73*	0.99	**1.5**
Min $\Delta u(k)$	*-1.06*	-1.07	**-2**
Pos. overshoot (m)	5.08	**5.89**	*0.5*
Vel. overshoot (m/s)	5.98	*3.3*	**6.24**
Transient at 5% (s)	**15**	*6*	10
# gear-switches	**6**	2	4
Effect of disturbances			
# violations	*0*	3	**6**
# violations (meas. dist.)	2	3	**27**
# violations (mdl. var.)	*0*	4	**6**
# gear-switches (dist.)	6	2	**38**

in the presence of disturbances. In particular two cases are reported: measurement disturbances (abbreviated *meas. dist.*) on position and velocity (uniformly random distributed error of $1\,m$ and $0.1\,m/s$ respectively) and model variation (abbreviated *mdl. var.*). In the latter case a scenario with wet asphalt (smaller friction coefficient $\mu = 0.005$), loaded vehicle (higher mass $m = 900\,kg$), long driving (higher tire pressure and bigger wheel radius $R = 0.30\,m$) is considered. The data presented with these experiments indicate the advantages and disadvantages of each method. In general it appears that the first method, that uses better approximated prediction model, gives less constraints violations, at the price of a higher computation effort. The highlight of the second method, offline, is the fast online computation time. On the contrary it may be unreliable in driving scenarios different than the nominal, like up-hill or down-hill driving, overloaded vehicle, different road conditions and so on.

V. CONCLUSIONS

The design of an adaptive cruise controller for a SMART is considered as a benchmark for a hybrid and linear MPC and for a PI approach. We have modeled the system in a significant level of details, including engine nonlinearities, hydraulic friction, tire/ground static friction and the gear shift. Constraints, arising from physical specifications were also considered. The control goal may be achieved by means of MPC-based techniques, for which we have used two versions (online and offline), with different prediction models. In

addition we have considered an ACC used in industry (based on a PI method). The comparison results on numerical and design issues are collected in a table, highlighting the main advantages and disadvantages of each method.

An interesting extension is to perform the comparison on a real field trial and driving cycles. We remark that, for a possible embedded solution in a real SMART, several technical issues should be regarded, like the sensor system, the resources of the on-board electronics, the real-life disturbances and the actuators delays. Furthermore it is relevant to investigate whether these results can be used for other applications and extended to a more general class of systems.

REFERENCES

[1] D. Swaroop and K. Rajagopal, "Intelligent cruise control and traffic flow stability," *Transportation Research Part C*, vol. 7, no. 6, pp. 329–352, Dec. 1999.

[2] V. Bageshwar, W. Garrard, and R. Rajamani, "Model predictive control of transitional maneuvers for adaptive cruise controller," *IEEE Trans. Vehicular Technology*, vol. 53, no. 5, pp. 1573–1585, Sep. 2004.

[3] D. Corona and B. De Schutter, "Adaptive cruise control for a SMART car: A comparison benchmark for MPC-PWA control methods," *IEEE Trans. Control Systems Technology*, vol. To appear, 2007.

[4] E. Sontag, "Nonlinear regulation: the piecewise affine approach," *IEEE Trans. Automatic Contr.*, vol. 26, no. 2, pp. 346–357, Apr. 1981.

[5] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, Mar. 1999.

[6] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming–the explicit solution," *IEEE Trans. Automatic Contr.*, vol. 47, no. 12, pp. 1974–1985, Dec. 2002.

[7] D. Yanakiev and I. Kanellakopoulos, "Nonlinear spacing policies for automated heavy-duty vehicles," *IEEE Trans. Vehicular Technology*, vol. 47, no. 4, pp. 1365–1377, Nov. 1998.

[8] D. Godbole and J. Lygeros, "Longitudinal control of the lead car of a platoon," *IEEE Trans. Vehicular Technology*, vol. 43, no. 4, pp. 1125–1135, Nov. 1994.

[9] S. Website, "http://www.smart-training-online.com/."

[10] F. Gustafsson, "Slip–based tire–road friction estimation," *Automatica*, vol. 33, no. 6, pp. 1087–1099, Jun. 1997.

[11] B. De Schutter and T. Van den Boom, "MPC for continuous piecewise-affine systems," *Systems & Contr. Letters*, vol. 52, no. 3–4, pp. 179–192, Jul. 2004.

[12] E. Kerrigan and D. Mayne, "Optimal control of constrained, piecewise affine systems with bounded disturbances," in *Proc. 41th IEEE Conf. on Dec. and Contr.*, Las Vegas, USA, Dec. 2002, pp. 1552–1557.

[13] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.

[14] R. Hallouzi, V. Verdult, H. Hellendoorn, and J. Ploeg, "Experimental evaluation of a co-operative driving set-up based on inter-vehicle communication," in *Proc. IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, Jul. 2004.

[15] M. Kvasnica, P. Grieder, M. Baotić, and F. Christophersen, *Multi-Parametric Toolbox MPT: User's Manual*, (ETH) Zurich, Jun. 2004, Available from http://control.ee.ethz.ch/~mpt.

[16] G. Marsden, M. McDonald, and M. Brackstone, "Towards an understanding of adaptive cruise control," *Transportation Research Part C*, vol. 9, no. 1, pp. 33–51, Feb. 2001.