

Low-complexity approximations of PWA functions: A case study on adaptive cruise control

A. Oliveri ^{*}, G.J.L. Naus [†], M. Storaice ^{*}, W.P.M.H. Heemels [†]

^{*} Biophysical and Electronic Engineering Department, University of Genoa, Italy

E-mail: {alberto.oliveri, marco.storaice}@unige.it

[†] Department of Mechanical Engineering, Technische Universiteit Eindhoven (TU/e), the Netherlands

E-mail: {g.j.l.naus, w.p.m.h.heemels}@tue.nl

Abstract—This paper applies recently developed techniques for the PieceWise-Affine (PWA) approximation of explicit Model Predictive Control (MPC) to an Adaptive Cruise Control system. The optimal MPC law is approximated by using a particular class of PWA functions defined over a domain partitioned into simplices, referred to as PieceWise-Affine Simplicial functions. This approximation technique allows a very fast circuit implementation of the control function, thereby enabling the usage of MPC in embedded systems with extremely small sampling periods.

I. INTRODUCTION

Model Predictive Control (MPC) is a popular technique for closed-loop control of multivariable systems subject to constraints on states, outputs and control inputs. At each sampling time, an optimization problem over a finite time horizon must be solved on line, in order to compute the optimal control move. The high computational effort of this approach can be overcome with so called explicit MPC [1], in which a multi-parametric optimization problem is solved offline. For linear systems, the resulting optimal control law is a PieceWise Affine (PWA) function of the state, defined over a domain partitioned into irregular polytopes. Although successfully applied in several practical applications [2], [3], [6] explicit MPC often generates a large set of polytopes, whose number depends roughly exponentially on the number of constraints included in the MPC optimization problem. For this reason the computation of the control function, especially for embedded digital controllers, can be prohibitive due to the complexity of finding the polytope containing the current state (point location problem).

Recently, many techniques have been studied to simplify or approximate the MPC controllers. In [4], for example, the MPC optimal law is approximated with a PWA function defined over a domain partitioned into *regular* polytopes called simplices. These special PWA functions, which are referred to as PWAS (PWA Simplicial), can be computed in a very efficient way by a digital embedded circuit due to the regular structure of the partition [5].

In this paper we apply the technique presented in [4] to the case study of the Adaptive Cruise Control (ACC) as described in [6]. ACC is an extension to standard cruise control functionality enabling to control both the throttle and the brakes

of a car (the host vehicle) in order to automatically follow a preceding vehicle (target vehicle) at a desired distance. ACC systems typically consist of a vehicle-independent part and a vehicle-dependent part [7], [8]. The first one determines the desired acceleration (deceleration) for the vehicle while the second part is responsible for the actuation of the throttle and brake system in order to track this desired acceleration (deceleration). In this paper, the vehicle-independent model presented in [6] is used to obtain an explicit MPC controller and an approximated PWAS control law with the technique presented in [4]. Our main purpose in the paper is to show the significant decrease that can be realized in computation time on a “real-life” example, without compromising the functionality of the controller. In addition, both the original explicit MPC and the approximated PWAS controller, have been implemented on a Spartan3AN FPGA and their performances have been evaluated by simulating three different realistic driving scenarios.

II. ACC MODEL

In this section, the ACC model is introduced, relying mainly on [6]. Assuming that the vehicle-dependent control part ensures perfect tracking of the desired host acceleration $a_{h,d}$, the internal vehicle dynamics and the vehicle-dependent control part together can be modeled by a single integrator $\dot{v}_h = a_{h,d}$, relating the host speed v_h to the desired acceleration $a_{h,d}$. The inter-vehicle relative distance x_r and relative speed $v_r = v_t - v_h = \dot{x}_r$ are measured by a radar installed on the host vehicle and measurements of the host speed (v_h) and acceleration (a_h) are available. The target acceleration a_t is unknown. Hence, it is, for now as a nominal case, assumed to be zero for the MPC prediction model, yielding $\dot{v}_r = a_r = -a_h$. In the end, a_t acts as a disturbance on the system. By discretizing the above mentioned equations with sampling period T_s , we obtain the following MPC prediction model:

$$\bar{x}_{k+1} = \begin{bmatrix} 1 & T_s & 0 & -\frac{1}{2}T_s^2 \\ 0 & 1 & 0 & -T_s \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{x}_k + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_k \quad (1)$$

where the state is $\bar{x}_k = [x_{r,k} \ v_{r,k} \ v_{h,k} \ a_{h,k}]'$ and the input u_k is the increment in the acceleration, defined as $u_k = a_{h,k+1} - a_{h,k}$. This formulation is necessary to impose constraints on jerk (derivative of the acceleration) which become constraints on the control input. Note that k indicates the discrete time and we have that $x_{r,k} = x_r(kT_s)$, $v_{r,k} = v_r(kT_s)$, $v_{h,k} = v_h(kT_s)$ and $a_{h,k} = a_h(kT_s)$, for $k \in \mathbb{N}$.

Typically, the objective of an ACC amounts to following a target vehicle at a desired distance $x_{r,d,k}$. Often, a so-called desired headway time $t_{hw,d}$ is used to define this desired distance, yielding

$$x_{r,d,k} = x_{r,0} + v_{h,k} t_{hw,d} \quad (2)$$

where $x_{r,0}$ is a constant term representing the desired distance at standstill. Correspondingly, the tracking error is defined as $e_k = x_{r,d,k} - x_{r,k}$.

The system (1) is subject to several inequality constraints on the state and input. To avoid collisions, x_r should always be positive. Furthermore, for comfort, the absolute value of the host acceleration and of the host jerk are constrained by $a_{h,min}$, $a_{h,max}$, $j_{h,min}$ and $j_{h,max}$. To further limit the state space that is explored when computing the MPC controller, a polytopic constraint is imposed whenever necessary, limiting the host and target speed by $v_{h,max} = v_{t,max}$ and the inter-vehicle distance by the radar range $x_{r,max} = x_{rr}$. Summarizing, the constraints are given by:

$$\mathcal{C}_1 : \begin{cases} 0 < x_r \leq x_{rr} \\ 0 \leq v_h \leq v_{h,max} \\ 0 \leq v_t \leq v_{t,max} \\ a_{h,min} \leq a_h \leq a_{h,max} \\ j_{h,min} \leq u \leq j_{h,max} \end{cases} \quad (3)$$

In order to make the system more suitable for the PWAS approximation, we recast the equations (1) as:

$$x_{k+1} = \begin{bmatrix} 1 & -T_s & 0 & T_s t_{hw,d} + \frac{1}{2} T_s^2 \\ 0 & 1 & 0 & -T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_k \quad (4)$$

with the new state variable $x_k = [e_k \ v_{r,k} \ v_{t,k} \ a_{h,k}]'$.

This new state variable leads to a rearrangement of the constraints (3) as follows:

$$\mathcal{C}_2 : \begin{cases} x_{r,0} + (v_t - v_r)t_{hw,d} - x_{rr} \leq e \leq x_{r,0} + (v_t - v_r)t_{hw,d} \\ 0 \leq v_t \leq v_{t,max} \\ v_t - v_{h,max} \leq v_r \leq v_t \\ a_{h,min} \leq a_h \leq a_{h,max} \\ j_{h,min} \leq u \leq j_{h,max} \end{cases} \quad (5)$$

III. EXPLICIT MPC CONTROLLER

The optimal control function $u^* = u^*(x_k) := U_{0,k}^*$ is obtained by solving the following optimization problem (multi-parametric program):

$$\min_{U_{0,k}^*, \dots, U_{N-1,k}^*} \sum_{l=0}^{N-1} [x_{l|k}^T Q x_{l|k} + U_{l,k}^{*T} R U_{l,k}^*], \quad (6)$$

where $x_{l|k}$ denotes the state predicted at time k for time $l+k$; clearly, $x_{0|k} = x_k$. Problem (6) is minimized according to (4) and (5) (for $l = 0, \dots, N-1$), where we chose a time horizon $N = 5$, weight matrices

$$Q = \begin{bmatrix} 2.5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 1 \quad (7)$$

and the following values for the constants in (5): $x_{r,0} = 3.5$, $x_{rr} = 200$, $t_{hw,d} = 1.5$, $v_{t,max} = 50$, $v_{h,max} = 50$, $a_{h,min} = -3$, $a_{h,max} = 2$, $j_{h,min} = -0.3$ and $j_{h,max} = 0.3$. The sampling time for the system is $T_s = 0.1$. We got an explicit solution u^* through the Matlab Multi-Parametric Toolbox [9]. The resulting control law $u^* = F_i x_k + G_i$, $x_k \in \mathcal{P}_i$ is a PWA function of the state x_k [1] defined on a four dimensional domain D_{MPC} partitioned into $N_p = 131$ polytopes \mathcal{P}_i , $i = 1, \dots, N_p$.

IV. APPROXIMATED PWAS CONTROLLER

The objective now is to obtain a PWAS approximation of the optimal MPC law $u^* : D_{MPC} \rightarrow \mathbb{R}$, which is much more convenient for a circuit-implementation point of view than u^* itself, but maintaining the favorable properties of u^* to some extent. Here we propose to use the approximation method of [4], which employs a particular class of PWA functions which are linear over regular polytopes called simplices. The simplicial partition is obtained by subdividing every dimensional component of a hyper-rectangular domain $S = \{x \in \mathbb{R}^n : x_{j,min} \leq x_j \leq x_{j,max}, j = 1, \dots, n\}^1$ into p_j sub-intervals of length l_{ji} , $i = 1, \dots, p_j$. Consequently, the domain is divided into $\prod_{j=1}^n p_j$ hyper-rectangles and contains $N_v = \prod_{j=1}^n (p_j + 1)$ vertices. Each hyper-rectangle is further partitioned into $n!$ simplices. If $l_{ji} = \frac{x_{j,max} - x_{j,min}}{p_j}$, $\forall j, i$, the partition is called uniform, otherwise it is referred to as non-uniform [10].

Each PWAS function can be expressed as a weighted sum of N_v PWAS basis functions, one for each vertex of the simplicial partition

$$f_{PWAS}(x) = \sum_{j=1}^{N_v} w_j \alpha_j(x) \quad (8)$$

Here $\alpha_j : S \rightarrow \mathbb{R}$ is a particular PWAS basis function, which takes the value 1 at the j -th vertex and 0 at all the others. Once the simplicial partition and the basis functions are defined, any PWAS function is completely defined by the weights w_k .

In order to find the best PWAS approximation of the optimal explicit MPC control law u^* , the distance (according to a specified norm) between the two functions is minimized. We decided to employ the 2-norm, which yields the approximating PWAS function \hat{u} by solving the optimization problem:

$$\min_{\hat{u}} \int_D [u^*(x) - \hat{u}(x)]^2 dx, \quad D = D_{MPC} \cap S \quad (9)$$

¹Here x_j denotes the j -th component of the state variable in a state-space representation, without reference to time.

By using (8), the functional (9) becomes a cost function of the weights $w = [w_1 \dots w_{N_v}]'$:

$$\min_w \int_D [u^*(x) - \sum_{j=1}^{N_v} w_j \alpha_j(x)]^2 dx, \quad (10)$$

which can be written as a quadratic program (QP)

$$\min_w w' H w - 2 f' w \quad (11)$$

where $H_{ij} = \int_D \alpha_i(x) \alpha_j(x) dx$ and $f_i = \int_D \alpha_i(x) u^*(x) dx$.

This QP is subject to equality and inequality constraints on w . The former force the approximated function to be 0 at $e = v_r = a_h = 0$. The latter impose the constraints (5) also on the PWAS control law. Due to the different partition structure of the optimal and approximated function, these last constraints could not be satisfied. Therefore, they are softened with a vector of slack variables σ , whose norm is minimized in the optimization problem. Constraints (5) can be recast as constraints on the weights w by imposing them only on the vertices of the partition obtained by combining the simplicial and the *irregular* partition of D_{MPC} . Due to the PWA nature of the control functions, this assumption ensures that the constraints are satisfied over the whole domain (see [4] for a more detailed discussion).

Hence, in conclusion, the weights of the approximated function are found by solving:

$$\min_{w, \sigma} w' H w - 2 f' w + \tau \sigma' \sigma \quad (12a)$$

$$s.t. \quad A_{eq} w = B_{eq} \quad (12b)$$

$$A w \leq B + \tau \sigma \quad (12c)$$

for suitable choice of A_{eq}, B_{eq}, A and B . We refer the reader to [4] for further details.

For the particular case study of ACC considered here, the simplicial partition is defined over a domain $S = \{x \in \mathbb{R}^4 : [-196 \ -35 \ 0 \ -3]' \leq x \leq [56 \ 35 \ 35 \ 2]'\}$, where the inequalities are intended componentwise; we use a non-uniform partition, with 15 segments along direction e , 14 segments along v_r , 1 segment along v_t and 15 segments along a_h . This implies that there are 75600 simplices and 7680 vertices.

V. RESULTS

We tested the performances of the approximated control function by simulating 3 different scenarios.

Scenario 1: the host vehicle is at a distance of 50 meters from the target vehicle and is proceeding at a constant speed of 30 Km/h (8.33 m/s). The target vehicle is at standstill. In this situation the host vehicle must brake and stop at a distance of 3.5 meters ($x_{r,0}$) from the target.

Scenario 2: the host vehicle is at a distance of 120 meters from the target vehicle and is proceeding at a constant speed of 40 Km/h (11.1 m/s). The target vehicle is moving at a constant speed of 70 Km/h (19.44 m/s). In this situation the host vehicle must reach the speed of 70 Km/h (19.44 m/s) and keep a distance of 32.66 meters from the target vehicle.

Scenario 3: the host vehicle is at a distance of 65 meters from the target vehicle and is proceeding at a constant speed of 110 Km/h (30.55 m/s). The target vehicle is moving at a constant speed of 70 Km/h (19.44 m/s). In this situation the host vehicle must reach the speed of 70 Km/h (19.44 m/s) and keep a distance of 32.66 meters from the target vehicle.

The time evolution of the system controlled with both the optimal and the approximated control law in the three scenarios described above is shown in Figures 1, 2 and 3, respectively. The monitored variables are the relative position x_r , the host speed v_h , the host acceleration a_h and the control output u . The gray solid lines are related to the optimal MPC control while the black solid lines correspond to the PWAS approximated control; the black dashed lines are the constraints to be fulfilled.

In all the scenarios the behavior of the system controlled with the approximated PWAS function is similar to the one with the MPC optimal controller: the host vehicle follows the target vehicle and keeps the desired distance from it.

The input constraints are always fulfilled while the state constraints are sometimes slightly violated due to the presence of the slack variables in (12).

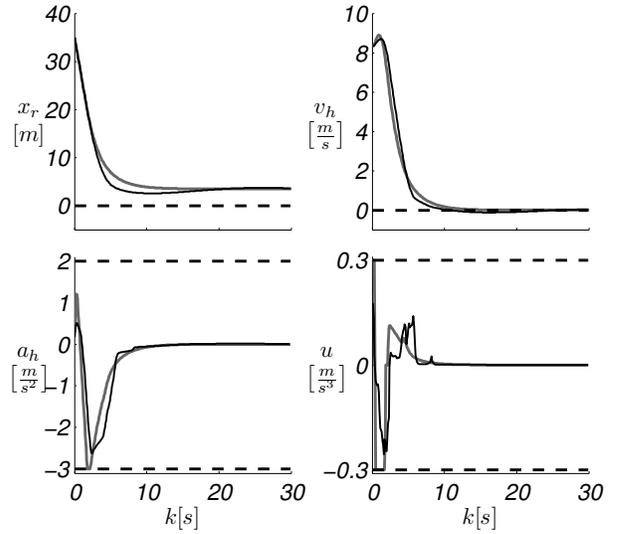


Fig. 1. Time evolution of the ACC system in scenario 1

VI. CIRCUIT IMPLEMENTATION

We implemented both the MPC optimal control u^* and the approximated control \hat{u} in a Xilinx Spartan3AN (XC3S700AN) FPGA. The MPC control function (of Section III) is implemented by using the architecture presented in [11], denoted as architecture A. Given a state x , the circuit finds the polytope \mathcal{P}_i containing it, by exploring a binary search tree. Once the polytope has been located, the control $u^*(x)$ is computed by evaluating the corresponding affine function $u^*(x) = F_i x + G_i$. The binary search tree is computed off-line and implemented in the circuit as a finite state machine.

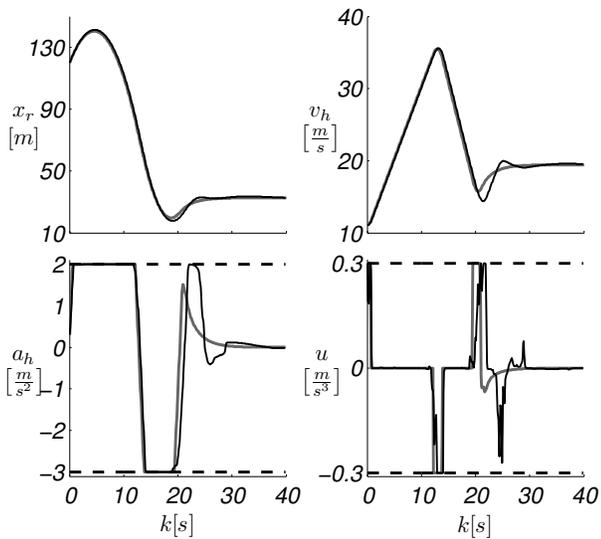


Fig. 2. Time evolution of the ACC system in scenario 2

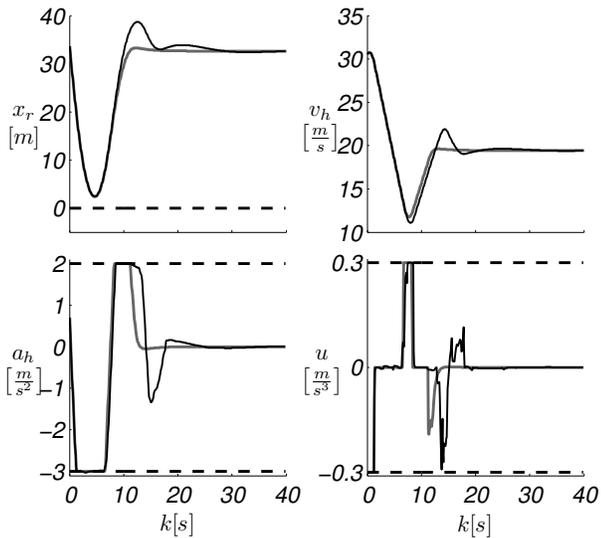


Fig. 3. Time evolution of the ACC system in scenario 3

In this application the binary search tree has 4495 nodes and a mean and maximum depth of 12.13 and 17, respectively.

The approximated PWAS function (of Section IV) is computed by using the two architectures proposed in [5], slightly modified in order to manage the non-uniform partition [10]. The simplicial partition has the advantage of drastically simplifying the point location problem (i.e., finding the simplex containing the state x), due to the regular structure of the simplices. The difference between the two architectures is that the former (architecture B) is serial (the operations are performed in sequence), while the latter (architecture C) is completely parallel. The serial architecture results in a slower but smaller circuit, while the parallel one achieves extremely small computation times, but occupies a larger area. The performances of the architectures are summarized in Table I.

| | % occup. slices | throughput | multipliers | memory size |
|---------|-----------------|-------------|-------------|-------------|
| Arch. A | 87% | 2.6 μ s | 1 | 3.3 KB |
| Arch. B | 31% | 390 ns | 1 | 11.5 KB |
| Arch. C | 95% | 67 ns | 5 | 57.6 KB |

TABLE I

Hence, this table shows that the approximate PWAS control laws result in a significant reduction in on-line evaluation times and circuit complexity (for Arch. B), while Section V shows that the control functionality and performance of the resulting ACC is comparable to the original high-complexity MPC law.

VII. CONCLUSIONS

We applied a recently developed technique of MPC approximation to the case study of an adaptive cruise control (ACC) system. The approximated PWAS controller has been implemented in a digital circuit whose performances have been compared to the performances of a circuit implementing the optimal MPC law. This showed that an enormous reduction in computational times can be realized, while the approximate PWA shows the same functionality as the original high-complexity MPC control law.

ACKNOWLEDGMENT

Work partially supported by the EC project MOBY-DIC “Model-based synthesis of digital electronic circuits for embedded control” (FP7-INFISO-ICT-248858), <http://www.mobydic-project.eu>.

REFERENCES

- [1] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, pp. 3-20, 2002.
- [2] S. Di Cairano, D. Yanakiev, A. Bemporad, I. Kolmanovsky, and D. Hrovat, “An MPC design flow for automotive control and applications to idle speed regulation,” in *Proc. 47th IEEE Conf. on Decision and Control*, Cancun, Mexico, 2008, pp. 5686-5691.
- [3] A. Beccuti, G. Papafotiou, R. Frasca, and M. Morari, “Explicit hybrid model predictive control of the DC-DC boost converter,” in *IEEE PESC*, pp.2503-2509, Orlando, Florida, USA, 2007.
- [4] A. Bemporad, A. Oliveri, T. Poggi, and M. Storace, “Ultra-Fast Stabilizing Model Predictive Control via Canonical Piecewise Affine Approximations,” *IEEE Transactions on Automatic Control*, in the press.
- [5] M. Storace, and T. Poggi, “Digital architectures realizing piecewise-linear multi-variate functions: two FPGA implementations,” *International Journal of Circuit Theory and Applications*, vol. 39, pp. 1-15, 2011.
- [6] G.J.L. Naus, J. Ploeg, M.J.G. Van de Molengraft, W.P.M.H. Heemels and M. Steinbuch, “Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach,” in *Control Engineering Practice*, vol. 18, pp. 882-892, 2010.
- [7] W. Prestl, T. Sauer, J. Steinle, and O. Tschernoster, “The BMW Active Cruise Control ACC”, *SAE Techn. Paper Ser.*, vol. 109, pp.119-125, 2000.
- [8] S. Moon, I. Moon, and K. Yi, “Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance,” *Control. Eng. Prac.*, vol. 27, pp. 442-455, 2009.
- [9] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari, “Multi-Parametric Toolbox (MPT)”, in *Hybrid Systems: Computation and Control*, vol. 2993, pp.121-124, 2004.
- [10] T. Poggi, F. Comaschi, M. Storace, “Digital circuit realization of piecewise-affine functions with nonuniform resolution: theory and FPGA implementation,” *IEEE Trans. Circ. Syst. II*, vol. 57, pp. 131-135, 2010.
- [11] A. Oliveri, A. Oliveri, T. Poggi, and M. Storace, “Circuit implementation of piecewise-affine functions based on a binary search tree,” *Proc. ECCTD09*, Antalya, Turkey, pp. 145-148, 2009.