2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

# Feasible Cooperation Based Model Predictive Control for Freeway Traffic Systems

J.R.D. Frejo and E.F. Camacho

***Abstract*- This paper proposes a distributed algorithm (Feasible Cooperation based Model Predictive Control, FC-MPC) for the control of freeway traffic systems. FC-MPC will be tested and compared with global, local and communicative MPC techniques in a traffic network of 18 segments with ITS (Intelligent Transport Systems) control signals: ramp meters and variable speed limits. It will be shown that local techniques have a suboptimal behavior and that centralized techniques are very difficult, if not impossible, to implement in real time. Communicative MPC improves the behavior of the controlled system versus the decentralized one. However, the solution is still suboptimal with respect to the centralized performance. On the other hand, FC-MPC is closed to the centralized behavior and has a much lower computational effort that the centralized one.**

Fig.1. Instant of 3D freeway traffic simulation with ITS signals (ramp metering and variable speed limits) using Aimsun 6.0

## I. INTRODUCTION

NOWADAYS, the fuel economy, the reduction of the atmospheric emissions and the reduction of the traffic accidents are important issues of the government policies in the first world. Freeway traffic causes a significant part of the CO2 emissions, fuel consumption and road accidents in advanced societies. Over the last years, much research has been focused on solving these problems. Since the construction of new freeways is not always a viable option, or it is not economic, other solutions are needed. In these cases, dynamic traffic control (the application of ITS control signals) may be a solution. Dynamic traffic control measure the state of the traffic (densities, velocities and queues) over time and computes control signals that change the response of the traffic system modifying its performance. Most useful control signals are "ramps metering" and "variable speed limits" (VSL) because they are easy to implement, relatively low cost and they allow a significant improvements in the Total Time Spent (TTS) by the drivers. Ramp metering and VSL have been already implemented successfully in USA, Germany, Spain, Netherlands and other countries [1] [2].

Nowadays, most of dynamic traffic control systems operate according to a linear and local control loop. However, the use of appropriate non-local and multivariable techniques will improve the reduction in the time spent by the drivers. Nonlinear centralized MPC is probably the best control algorithm choice for a small network as can be seen on [3]. The main problem of nonlinear centralized MPC is that the computational time quickly increases with the size of the network. Thus, centralized MPC could be impossible to apply for large networks. A possible solution is to consider the network as a set of subsystems controlling each subsystem by one independent MPC (i.e. to use a decentralized control scheme).

The main objective of this paper is to design a control algorithm that can be implemented in real time for a large enough traffic network minimizing the total time spent by the drivers. In other to test its performance, the proposed algorithm will be compared with other possible solutions (decentralized MPC, centralized MPC, and communicative MPC) in a simulated freeway.

The research leadind to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement nº257462 HYCON2 Network of excellence.

The authors are with the Dept. de Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros, University of Seville, Spain jdominguez3@us.es eduardo@esi.us.es

## II. METANET MODEL

The model used for the design of the controller is the macroscopic traffic flow model METANET [4]. The METANET model is a second-order model that is discrete in both space and time. The METANET model represents a network as a graph where the links *(m)* corresponds to freeway stretches. Each link *m* is divided into *Nm* segments (*i*) of length *Lm.* Each segment is characterized dynamically by the *traffic density* $\rho_{m,i}(k)$and the *mean speed* $v_{m,i}(k)$ where *k* is the instant *t=kT*. *T* is the simulation time step (10 s). $q_{m,i}(k)$ is the *traffic flow* which can be computed for each time step using:

$$q_{m,i}(k) = \rho_{m,i}(k) v_{m,i}(k)\, \lambda_m \quad (1)$$

The system dynamics are described by two equations. The first one expresses the conservation of vehicles:

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}\left(q_{m,i-1}(k) - q_{m,i}(k)\right) \quad (2)$$

The second one expresses the mean speed as a sum of the previous mean speed (3), a relaxation term (4), a convection term (5) and an anticipation term (6):

$$v_{m,i}(k+1) = v_{m,i}(k) \; + \tag{3}$$
$$+ \frac{T}{\tau}\left(V\left(\rho_{m,i}(k)\right) - v_{m,i}(k)\right) + \tag{4}$$
$$+ \frac{T}{L_m} v_{m,i}(k)\left(v_{m,i-1}(k) - v_{m,i}(k)\right) - \tag{5}$$
$$- \frac{\eta T}{\tau L_m} * \left(\rho_{m,i+1}(k) - \rho_{m,i}(k)\right) / \left(\rho_{m,i}(k) + K\right) \tag{6}$$

Where $V\left(\rho_{m,i}(k)\right)$ is the desired velocity of the drivers. The desired velocity models the static characteristic of the traffic system. In this equation appears the effect of the VSLs in the control variable $v_{control,m,i}(k)$:

$$V\left(\rho_{m,i}(k)\right) = \min(v_{free,m} \exp\left(-\frac{1}{a_m} * \left(\frac{\rho_{m,i}(k)}{\rho_{crit,m}}\right)^{a_m}\right), \quad (1{+}\propto) v_{control,m,i}(k)) \tag{7}$$

In order to compute the flow that enters from a ramp metering the following equation is used:

$$q_o(k) = \min(\; (d_o(k) + \frac{w_o(k)}{T}), \; C_o r_o(k), C_o(\rho_{max} - \rho_{m,1}(k))/(\rho_{max} - \rho_{crit,m})) \tag{8}$$

Where $d_o(k)$ is the ramp flow demand, $r_o(k)$ is the control variable of the ramp metering and $w_o(k)$ is the queue of the ramp. The queue is a new dynamic variable characterized by $w_o(k+1) = w_o(k) + T(d_o(k) - q_o(k))$ Moreover, when a ramp enters in a freeway a penalization term has to be added into velocity equation (3):

$$-\delta T q_o(k) v_{m,1}(k) / L_m \lambda_m (\rho_{m,1}(k) + K) \tag{9}$$

For the first segment of the network, it has been considered than the upstream velocity $v_{m,i-1}(k)$ is equal to the current velocity $v_{m,i}(k)$. The same consideration is used for the density downstream the last segment.

## III. MODEL PREDICTIVE CONTROL (MPC) IN TRAFFIC CONTROL SYSTEMS

### A. Introduction

Model Predictive Control [5] is a flexible approach towards the dynamic traffic control problem that optimizes a cost function using a model in a receding horizon framework. In traffic control, the cost function minimizes the total time spent by all the drivers (or other performance o safety criteria) and the model used to be a macroscopic traffic model like METANET. By merely changing the cost function, the implemented policy can be changed. Moreover, MPC can take constraint into account and deal with slow changes in the behavior of the traffic systems.

### B. Quick description of model predictive control

MPC originated in the late seventies and has been developed considerably since them. The main ideas of model predictive control are basically:

--Explicit use of a model to predict the process output at future time instants (prediction horizon)

--Calculation of a control sequence minimizing an objective function

--Receding strategy, so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence computed at each step.

The various MPC algorithms only differ amongst themselves in the model used to represent the process and the cost function to be minimized.

The main advantages of MPC are that it is very intuitive (during the design and the tuning), it can deal with complex model (for example, nor linear or multivariable models), it has compensation for dead times intrinsically, it can use future references…

The main disadvantage is the computational time needed, especially for nor-linear multivariable cases.

### C. Previous works on MPC for traffic systems

MPC have been successfully tested in simulations in traffic systems. In [6], 2 simulations using ramp metering with ALINEA, most implemented local control algorithm in the computation of the ramp metering rates, or MPC control algorithm are compared obtaining a decrease of 1.3% in the ALINEA case and 6.9% in the MPC case. In [7], VSL are previously determined without an optimization of a macroscopic model (taking account of factors as maximize a bottleneck flow, the limits on queues lengths…). After this, the ramps metering are computed using MPC. In [8], Ramp metering rates are computed previously with a given strategy (for example, ALINEA). Following, VSL are calculated using MPC with a simplified METANET model. Using this algorithm, a reduction of the 31,8 % in the TTS is obtained in a simulation for a real network. In [3], it is demonstrated that the use of speed limits in a MPC control framework for traffic systems with ramps metering and VSLs can substantially improve the network performance. The improvement of the network simulated in the TTS is a 14.3% being just a 5.3% if only ramps metering are used. In [9], a comparison between decentralized and centralized MPC algorithm is done concluding that the decentralized options are quite suboptimal (6.5 % against 26,4 %) and that the centralized MPC is impossible to implement in real time for a large traffic network.

It is important to note that the reduction of the TTS strongly depends on the traffic condition. In order to properly compare two algorithms, they must be simulated in the same network and conditions.

### D. Distributed MPC

Most standard model predictive control implementations divide the system into several parts and apply MPC individually to the units. It is know that such a completely decentralized control strategy may result in unacceptable control performance, especially if the units interact strongly as in control traffic systems. Completely centralized control of large networks is viewed by most practitioners as impractical and unrealistic.

Distributed MPC algorithms [10] try to solve the problem in a parallel computation using the communication and cooperation between the different MPC controllers in order to achieve the centralized performance.

In communicative MPC, the interactions between systems are modeled in order to that each controller takes account of the actions of its neighbors. In each interaction, the predicted trajectories are exchanged between controllers and the optimization is repeated with the new values of the control signal profiles of its neighbors. If the algorithm converges (proved for linear systems), the "Nash Equilibrium" is reached. However, the "Nash Equilibrium" is suboptimal for many systems such as the traffic system.

In order to improve the behavior, cooperative MPC can be used. This technique modifies the objective functions of the local MPCs including also the objective functions of near agents properly weighted. The iterations and the exchange of information are done in the same way that communicative MPC. In feasible cooperation based MPC (FC-MPC), only the local variables corresponding to each controller are used as decision variables. In [10], it is proven that FC-MPC converges to the optimal centralized MPC control (Pareto-optimum). There are no proven results for non-linear cases. However, in this paper is possible to see how for traffic systems the centralized MPC behavior can be roughly reached using FC-MPC.

## IV. BENCHMARK

### A. Benchmark

In order to analyze the different controllers designed (on following sections) the network of Fig.2 has been analyzed.

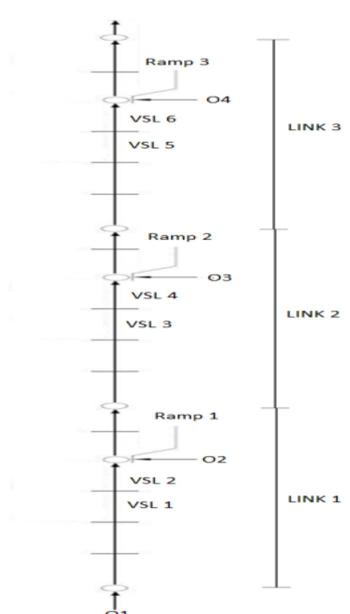


Fig.2. Traffic Network simulated.

The network is an 18 kms freeway with 3 Ramp Metering and 6 Variables Speed Limits. The freeway is discretized in 18 segments of 1000 meters ($L$) with 2 lanes ($\lambda$). All the model parameters are considered equals for all the segments. The remaining model parameters are as follow:

$\tau = 0.005\ hour$, $K = 40\frac{veh}{Km*Line}$, $\rho_{crit} = 33.5\frac{veh}{Km*Line}$, $a = 1.867$, $v_{free} = 102\frac{Km}{hour}$, $\eta = 60\frac{Km*Km}{hour}$, $\delta = 0.0122$, $\rho_{max} = 180\ \frac{veh}{Km*Line}$, $C_o = 4000\frac{veh}{hour}$, $C_{ramp} = 2000\frac{veh}{hour}$ and $\propto = 0.1$.

The input flow demands are chosen in order to obtain a simulation with a high density where the traffic control can improve substantially the behavior of the system. The simulation time chosen is two and half hour that corresponds to 75 controller sample time and 900 simulation steps.

In a small network, a centralized MPC could be computed in less time than the controller sample time allowing implementing the control signal in real time. However, the network analyzed is big enough to make impracticable a centralized controller.

Since there is only one destination, the biggest traffic density will appear in the last link. The control actions in links 1 and 2 will have a large effect in the third link that could increase the traffic jam in this link. Therefore, in this network (as happens in real traffic networks), the consideration of the effects of the neighboring controllers will be a critical issue.

## V. LOCAL AND GLOBAL MPC

### A. Local MPC

In order to design a local MPC controller, just one link has been considered as the full network. In this case, the three links have the same structure and, therefore, only will be necessary to design one controller which will be used in each link. The geometry of each link can be seen on Fig. 3.

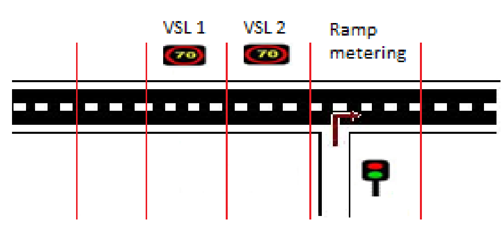


Fig.3. Stretch used in the local MPC

Since there is no communication between controllers, the future disturbances (estimation of upstream velocity and flow and downstream density) are defined by the simulation of the no-control case for any of the agents, which is obviously unreal. There are 3 control signals: VSLs on segments 3 and 4 and a ramp metering in segment 5. Thirteen variables are measured at each sample time (mean density and velocity of each segment and queue of the ramp) and used for the computation of the control signals.

In this section, the main particularities of the MPC controller designed are explained (All the aspects which are not explained here have the general structure of a nonlinear MPC as in [5]):

--The controlled system is subject to constraints in the maximum and minimum values of density, velocity, queue, control velocities of the VSLs and ramp metering rate.

--The cost function used by the controller is the following:

$$J(k_c) =$$

$$\sum_{k=Mk_c}^{M(k_c+N_p)-1}[T\sum_{i=1}^{i=6}(\rho_i(k)\,L\lambda)+w_o(k)] \quad (10)$$

$$+\ \sum_{l=k_c+1}^{k_c+N_c}\varepsilon\|u(l|k_c)-u(l-1|k_c)\|^2 \quad (11)$$

$$+\ \varepsilon\|u(k_c|k_c)-u(k_c-1|k_c)\|^2 \quad (12)$$

Where $k_c$ is the control step time $t=k_cT_c$ with $T_c$=120 s. The controller step time must be higher (in this case, 12 times higher) that the simulation step time in order to have enough time for the computation of the optimization. The first term (10) of the cost function expresses the total time spent (TTS) by all the drivers during the prediction horizon. The second term (11) and the third term (12) express the variations of the control signals.

--The function "fmincon" of the Optimization Toolbox of Matlab have been used in order to compute the optimization. This function uses SQP optimization techniques. If all the variables (16*Np) with its respective constraints are considered, the optimization cannot be computed in a reasonable time. For this reason, only the control variables are included in the optimization explicitly. The constraints in velocity, density and queue are made soft including penalization terms in the cost function.

--In order to try to avoid that the algorithm falls in a local minimum, an "evaluation procedure" is run before the optimization. During it, the TTS is evaluated for a grid of control values. The minimum value obtained in this evaluation is taken as initial value for the optimization.

During the tuning of the controller, all the cost parameters have been set in order to obtain the minimum TTS. The results are very sensitive with the tuning and, therefore, a meticulous tuning procedure has to be done for each network. Especially important are the set of $\varepsilon$ (i.e. the parameters that multiply the penalization in the changes in the control signals).

In theory, the penalization factors which multiply the soft constraints of density, velocity and queues have to be large. However, in practice, these factors cannot be too large or numerical problem will appear during the optimization.

An increase in the horizons will improve the behavior and, at the same time, will increase the computational time needed. A good trade-off between cost and behavior is to choose the prediction horizon between 3 and 7. In general, the horizons size will depend on the size of the network. For a large network, more decrease in the TTS will be obtained increasing the horizons but more critical will be the computational time. It is important to note that the difference between the control horizon and the prediction horizon has to be small or 0, in order to obtain a good behavior. It happens because does not make sense to consider constant the control input during a long final period due to the system does not tend to an equilibrium point. If we set an *(Np – Nc)* too large, the system takes too much into account the final values of the control signal causing a suboptimal behavior. In this paper, *Np = Nc = 3* for all the controllers.

### B. Centralized (global) MPC

The centralized MPC is a controller that optimizes the full network (18 kms) for a given prediction and control horizons. It has the same structure that local MPC but increasing the size of the network (i.e. the number of variables and constraints). The behavior of the network must be better or equal than any nor-centralized controller.

The main problem of this controller is that the number of decision variables is increased critically. In this case, the full network has 9 variables that need to be implemented at each time. Taking account of the control horizon, the optimization problem will need to find *9*Nc* decision variables. Since the METANET model has 2 (or 3 in ramps) non-linear equations for each segment and for each time, the computational time needed will increase critically with the size of the network. In practice, a centralized MPC for a large scale traffic network will be impossible to implement in real time.

## VI. DISTRIBUTED MPC SOLUTIONS

### A. Local MPC with Communication after Sample

As can be seen on equation (2) and equation terms (5) and (6), the upstream flow and velocity and the downstream density is necessary in order to model a segment. Therefore, each MPC controller will need the current and future values of these variables. These variables can be seen as estimable disturbances. The communication between controllers after any sample will allow to local MPC to use an estimation of these disturbances that are defined by the predicted values of the adjacent MPC controllers (Fig. 4).

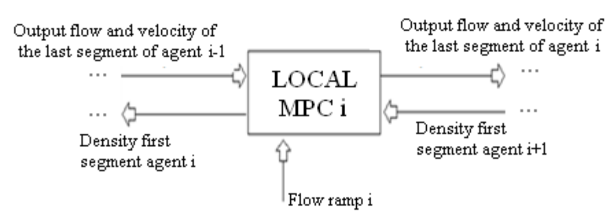


Fig.4. Controller interconnection structure. Where the variables are sent between controllers, also the predicted values of them during the prediction horizon are sent.

After any sample time, a controller will send to the previous controller the future predicted values of the density of its first segment and to the following controller will send the future predicted values of its output flows and of the velocity of the last segment. It will allow the others controllers to use a more real prediction of the disturbances (input flow and downstream density). However, the controllers will not take into account the effects of their acts in the others parts of the networks. Thus, it can be said that this controller communicate but not cooperate with their neighbors.

### *B. Communicative MPC*

Communicative MPC uses the same procedure that the "Local MPC with Communication after Sample" but doing the process "communication + optimization" many times inside a controller step time. In this paper, 4 iterations were used and it was enough in the majority of the cases to converge to an equilibrium point, which is the Nash Equilibrium. This control technique has the same problem explained previously: the controllers do not cooperate and, therefore, two controllers could be counteracting.

### *C. Cooperative MPC (Feasible Cooperation Based MPC)*

In the cooperative MPC the local cost function of each controller is replaced by a global cost function (in this case, the TTS of the 18 segments). In order to reduce the computation effort, only the control signals of each part of the network (in this case, two VSLs and one ramp metering) are considered as decision variables. Therefore, the following cost function is considered:

$$J(k_c) = \sum_{k=Mk_c}^{M(k_c+H_p)-1}\left[T\sum_{(m,i)\in I}(\rho_{m,i}(k)\, L_m\lambda_m + \sum_{o\in O} w_o(k)\right] \quad (13)$$
$$+\sum_{l=k_c+1}^{k_c+H_c}\varepsilon\|u(l|k_c) - (l-1|k_c)\|^2 \quad (14)$$
$$+\,\varepsilon\|u(k_c|k_c) - u(k_c-1|k_c)\|^2] \quad (15)$$

Where $I$ is the full network (in a bigger network, $I$ would be the part of the network corresponding to the controller adding some upstream and downstream parts) and $O$ is the set of all origins. The first term of the cost function (13) considers the TTS of the full network (18 segments) and the second (14) and third term (15) expresses the variables of the control signals considered in each controller. In Cooperative MPC, all the variables contained in $I$ could be optimized. However, it will increase the computational time since we are solving the centralized MPC in each iteration. FC-MPC tries to solve this problem reducing the decision variables to the local variables of the controller. Therefore, the optimization variables are:

$$u(l|k_c) = [v_{control1,m_{cont}}(l|k_c), v_{control2,m_{cont}}(l|k_c), r_{m_{cont}}(l|k_c)] \quad (16)$$

In this simulation, FC-MPC uses only 4 iterations as Communicative MPC. In the majority of the sample times, it is enough to converge. It is important to note that the computational time required for each iteration rapidly decrease. It happens because we start each optimization in the optimum obtained in the previous iteration. As the optimization problem changes just a little bit when the variables are exchange, the optimization algorithm does not need to move to a very far point.

## VII. CONTROLLERS SUMMARY AND RESULTS

### *A. Summary of the cost functions and decision variables*

The various controllers use different cost function: $J_i$ is the cost function associated to link $i$ and $J$ is the cost function associated to the full network (see Fig.5).

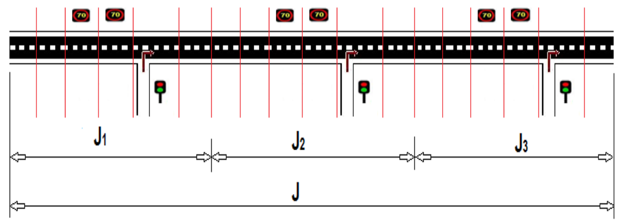


Fig.5. Local and Global TTS cost functions in the network.

In Table I the cost function and the decision variables used by each controller are summarized. The term $u_{i,past}$ expresses that the controllers are using the past values of the control variables of the link $i$ but not the value that will be implemented in the current sample time.

TABLE I
SUMMARY OF THE DIFFERENT CONTROLLERS

| | Controller 1 | Controller 2 | Controller 3 |
|---|---|---|---|
| *Local MPC* | $\min_{u_1} J_1(u_1)$ | $\min_{u_2} J_2(u_2)$ | $\min_{u_3} J_3(u_3)$ |
| *Local MPC with communication after sample* | $\min_{u_1} J_1(u_1, u_{2,past}, u_{3,past})$ | $\min_{u_2} J_2(u_2, u_{1,past}, u_{3,past})$ | $\min_{u_3} J_3(u_3, u_{1,past}, u_{2,past})$ |
| *Communicative MPC* | $\min_{u_1} J_1(u_1, u_2, u_3)$ | $\min_{u_2} J_2(u_1, u_2, u_3)$ | $\min_{u_3} J_3(u_1, u_2, u_3)$ |
| *FC-MPC* | $\min_{u_1} J(u_1, u_2, u_3)$ | $\min_{u_2} J(u_1, u_2, u_3)$ | $\min_{u_3} J(u_1, u_2, u_3)$ |
| *Centralized MPC* | $\min_{u_1,u_2,u_3} J(u_1, u_2, u_3)$ | | |

### *B. Results*

The controllers have been simulated using Aimsun 6.0 software (see Fig. 6) supposing that the system response as a METANET model. In this program, it is also possible to use a microscopic model for the response of the system. However, it would be necessary a long calibration procedure and it is not clear that a microscopic model obtains more realistic results than a macroscopic model.

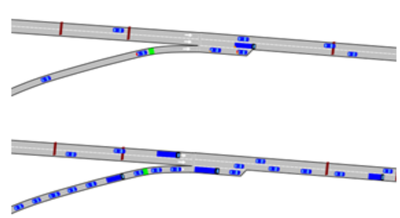

Fig.6. Instant of 2D Simulation with Aimsun 6.0 of the controlled (using FC-MPC) and uncontrolled cases.

The results of Table II have been obtained assuming that the real system responses according to the macroscopic METANET model used as control model. As can be seen on results, all the controllers reduce the TTS (*Red* is the reduction of the TTS in percentage with respect to the non-linear case). At the same time, all the controllers keep the variables inside the constraints. In the table, *MCT* shows the maximum computation time (in the local case, *MCT* have been taken from the worst cases, i.e. more restrictive cases, of the 3 controllers).

TABLE II
RESULTS OF THE IMPLEMENTATION OF THE DIFFERENT CONTROLLERS

| | Hc | Hp | TTS | Red | MCT |
|---|---|---|---|---|---|
| *Uncontrolled system* | -- | -- | 1684 | 0 | -- |
| *Local MPC without communication* | 3 | 3 | 1573.6 | 6.5 | 51.6 |
| *Local MPC with communication after sample* | 3 | 3 | 1493.4 | 11.3 | 29.1 |
| *Communicative MPC* | 3 | 3 | 1455.8 | 13.6 | 45.1 |
| *Cooperative MPC (FC-MPC)* | 3 | 3 | 1262 | 25.1 | 55.1 |
| | 5 | 7 | 1247.2 | 25.9 | 237.7 |
| *Centralized MPC* | 3 | 3 | 1252.8 | 25.6 | 231.5 |
| | 5 | 7 | 1238.8 | 26.4 | 1103.9 |

Analyzing the results of the local MPC, it can be seen how the communication after sample increase substantially the reduction of the TTS (from 6.5% to 11.31%).

If the communication is done inside the sample time the reduction in the TTS increase just from 11.31 % to 13.55 %. However, if cooperation is considered (using FC-MPC) the reduction achieves the 25.06 % (really close to the 25.6 % of the centralized MPC). It shows that the Nash Equilibrium is quite suboptimal in freeway traffic systems using METANET model. The reason is that a good traffic control system needs to the take into account the effects of the own traffic control system in other parts of the network. Without this consideration, solving a traffic jam in one part of the network could increase the number of vehicles that arrive to a bigger traffic jam, getting worse the global behavior of the network.

On the other hand, it can be seen how the FC-MPC approximates the centralized behavior in just a few iterations (4 in this case) showing the Pareto-Optimum can be reached.

Looking the computational time needed it is possible to see how the centralized controller cannot be implemented in real time for all sample even with a small horizons such as 3. However, decentralized computations requires between a third and fifth of the computational effort making them implementable in real time. For higher horizons (5-7), the reduction of the TTS is increased a small percentage but the computational time is highly increased.

It is important to note that the minimization of the TTS is just a criterion for the operation of the traffic system. Other objectives can be considered as the reduction of emission, the homogenization of the traffic flows or the minimization of fuel consumption. This is one of main advantages of Model Predictive Control; the policy can be changed just modifying the cost function without changing any other part of the controller.

## VIII. CONCLUSION

In this paper, an 18 km freeway has been simulated for some different control techniques. The first conclusion is that a fully decentralized controller for a traffic network is quite suboptimal (6.5 % against 26.4 % in the reduction of the TTS) and that centralized NLMPC is not implementable in real traffic network due to the computational effort needed. The second is that a distributed MPC algorithms converge, in this case, in just a few iterations and can be computed in a fraction of the time needed by the centralized one (between a third and a fifth). The third conclusion is that Nash Equilibrium is far away to Pareto Optimal Equilibrium in this traffic system (i.e. cooperation is a key issue). In the simulation, the TTS decreases from 11.31% to 25.06% thanks to cooperation. It allows that FC-MPC almost reaches the centralized behavior fulfilling the design objectives. Moreover, as was shown in previous papers, it can be seen how the ITS signals substantially improves the performance of the traffic system, especially if the control signals are computed using model predictive control.

## REFERENCES


[1] Minnesota Department of Transportation. “Ramp Meter Study” 2001

[2] E. van den Hoogen and S. Smulders. “Control by variable speed signs: results of the Dutch experiment” in *Road Traffic Monitoring and Control, 1994.Seventh International Conference on,* pages 145-149, 26-28 Apr 1994

[3] A. Hegyi, “Model Predictive Control for Integrating Traffic Control Measures”. *Technische Universiteit Delft. Feb 2004*

[4] A. Messmer, M. Papageorgiou. “METANET: A macroscopic simulation program for motorways networks” in *Traffic Engineering and Control, 31.*pages 466-470. 1990

[5] E.F. Camacho and C. Bordons “Model Predictive Control”. Springer. Second Edition 2004.

[6] T. Bellemans, B. De Schutter, B. De Moor “Model Predictive Control for ramp metering of motorway traffic: A case study” in *Control Engineering Practice 14 on* pages 757-767. 2006

[7] X. Lu, P. Varaiya, R. Horowitz, D. Su and S.E .Shladover “A New Approach for Combined Freeway Variable Speed Limits and Coordinated Ramp Metering” in *13th Annual Conference on Intelligent Transportation Systems,* pages 19-22. September 2010

[8] X. Lu, T. Z. Qiu, P. Varaiya, R. Horowitz and S.E. Shladover “Combining Variable Speed Limits with Ramp Metering for Freeway Traffic Control” in *American Control Conference 2010.* July 2010

[9] J.R.D. Frejo and E.F. Camacho “Global Versus Local MPC Algorithms in Freeways Traffic Control with Ramp Metering and Variable Speed Limits”. Submitted for *IEEE Transactions on Intelligent Transportation Systems* 2011

[10] A.N. Venkat, “Distributed Model Predictive Control: Theory and Applications”. Ph.D. University of Wisconsin-Madison 2006