Delft University of Technology

Distributed MPC for Large Freeway Networks Using Alternating Optimization

Todorović, Uglješa; Frejo, José Ramón D.; Schutter, Bart De

DOI
10.1109/TITS.2020.3028850

Publication date
2022

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

Citation (APA)
Todorović, U., Frejo, J. R. D., & Schutter, B. D. (2022). Distributed MPC for Large Freeway Networks Using Alternating Optimization. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 1875-1884. https://doi.org/10.1109/TITS.2020.3028850

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher
is the copyright holder of this work and the author uses the
Dutch legislation to make this work public.

Distributed MPC for Large Freeway Networks Using Alternating Optimization

Uglješa Todorović, José Ramón D. Frejo, and Bart De Schutter, *Fellow, IEEE*

***Abstract*— The Model Predictive Control (MPC) framework has shown great potential for the control of Variable Speed Limits (VSLs) and Ramp Metering (RM) installations. However, the implementation to large freeway networks remains challenging. One major reason is that, by considering the VSLs to be discrete decision variables, an extremely difficult Mixed Integer Nonlinear Programming (MINLP) optimization problem has to be solved within every controller sampling interval. Consequently, many related papers relax the MINLP problems by considering the VSLs to be continuous variables. This paper proposes two novel MPC algorithms for coordinated control of discrete VSLs and continuous RM rates that do not make this relaxation. The proposed algorithms use a distributed control architecture and an alternating optimization scheme to relax the MINLP optimization problems but still consider the VSLs as discrete variables and, hence, offer a trade-off between computational complexity and system performance. The performance of the proposed algorithms is evaluated in a case study. The case study shows that relaxing the VSLs to be continuous variables with a distributed architecture results in a significant performance loss. Furthermore, both proposed algorithms have a lower computational complexity than the more conventional centralized approach and, as a result, they do manage to solve all optimization problems within the sampling intervals. Moreover, one of the proposed algorithms has a system performance that is remarkably similar to the optimal performance of the centralized approach.**

***Index Terms*— Alternating optimization, distributed MPC, freeway traffic control, ramp metering, variable speed limits.**

I. Introduction

As THE number of vehicles and the need for transportation keeps increasing every year, traffic congestion has become a crucial problem in today's society. There is a need for a sustainable solution to reduce or even eliminate traffic jams. Freeway traffic control has shown to be a sustainable solution to this problem [1]–[3].

Especially the implementation of Ramp Metering (RM) installations and Variable Speed Limits (VSLs) as control measures is currently a widely researched area, because proper coordination of those measures has shown great potential for the reduction of traffic congestion, traffic emissions, and the risk of accidents [4], [5]. However, for optimal performance, these control measures have to be coordinated at a large spatial scale as local control inputs have significant influence on the traffic state of distant parts of the freeway and, therefore, on the global network performance [6].

The Model Predictive Control (MPC) [7] framework has shown outstanding potential for proper coordination of those control measures [4], [5], [8], [9], but the real-life implementation to large freeway networks remains challenging. One major challenge, on which the current paper focuses, is that often extremely difficult optimization problems have to be solved for the application of MPC to large freeway networks. Some other challenges, which are not addressed by the current paper but are still a topic of ongoing research, include: robustness analyses of MPC for freeway control [10], validation of the prediction models with empirical data [11], and the prediction of future disturbances.

The difficult optimization problems often arise for the following reasons. Firstly, due to the inherent nonlinearity of traffic flow, mostly nonlinear prediction models are used in literature [4], [12], [13]. Moreover, by law, VSLs are only allowed to take prescribed discrete values, while RM rates are typically continuous control inputs (which are usually afterwards translated into traffic signal cycles [14]). The combination of a nonlinear prediction model with discrete VSLs and continuous RM rates yields a Mixed Integer Nonlinear Programming (MINLP) optimization problem that has to be solved within every controller sampling interval. In general, the computation time that is needed to solve such an optimization problem increases exponentially with the size of the problem (i.e. with the number of control measures, the control horizon, etc.). Hence, for the application of MPC to large freeway networks, the computation time quickly becomes larger than the controller sampling time, making MPC unimplementable.

Various approaches have been considered to reduce the complexity of the optimization problems. Some approaches simplify the nonlinear prediction model by rewriting it as a mixed logical dynamical model [15], relaxing the MINLP optimization problems into mixed integer linear programming optimization problems [16], [17]. Other approaches parametrize the control signals into control laws to simplify the optimization problems [18]. Some approaches consider the VSLs as continuous decision variables [4], [5], relaxing the MINLP optimization problems into nonlinear continuous optimization problems. However, related papers draw contradictory conclusions on this relaxation, as some have found that it results in a large performance loss [19], while others report that the performance loss is not significant [20]. Hence, the effect of

Manuscript received January 21, 2020; revised July 19, 2020 and September 28, 2020; accepted September 30, 2020. Date of publication October 9, 2020; date of current version March 9, 2022. The Associate Editor for this article was I. Papamichail. *(Corresponding author: Uglješa Todorović.)*

Uglješa Todorović and Bart De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: uggi.todorovic@gmail.com; b.deschutter@tudelft.nl).

José Ramón D. Frejo is with the Department of Systems Engineering and Automation, University of Seville, 41004 Seville, Spain (e-mail: jdominguez3@us.es).

Digital Object Identifier 10.1109/TITS.2020.3028850

1558-0016 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

this relaxation on the system performance is still an open question in the field and it is investigated in the current paper.

A distributed control architecture [21] is a popular approach to reduce the computational complexity of large-scale MPC problems in general. However, the implementation of such an architecture to freeway networks is difficult, as every distributed agent still has to solve an MINLP problem. Hence, the existing distributed approaches with application to freeway traffic control either do not consider VSLs as control inputs [16], [22], [23], or consider these to be continuous decision variables in the optimization [5], [6].

The main contribution of the current paper is the proposal of two novel MPC algorithms that use the distributed control architectures proposed in [5] and [23], and an alternating optimization scheme similar to [19] for coordinated control of discrete VSLs and continuous RM rates: Fully Cooperative Alternating Model Predictive Control (FC-A-MPC) and Downstream Cooperative Alternating Model Predictive Control (DC-A-MPC). Both proposed algorithms are similar, but offer a different trade-off between computational complexity and system performance. Moreover, both algorithms reduce the computational complexity of the problem significantly, such that they are implementable in real time for large freeway networks. Furthermore, this paper contributes to the state-of-the-art by investigating the effects on the system performance by relaxing the VSLs to be continuous decision variables in a distributed setting.

The remainder of this paper is structured as follows. Section II outlines the components of the MPC framework that are used in this work. Subsequently, Section III proposes the two novel MPC algorithms. Then, Section IV presents a case study that evaluates the proposed algorithms. Finally, Section V concludes this paper.

II. MPC for Freeway Traffic

A. Introduction

MPC is an advanced control methodology where an objective function is optimized to find optimal control inputs and a prediction model is used to predict relevant future system trajectories. With the optimization, multiple control inputs can be determined, and the prediction model can be used to predict influences of those control inputs on distant parts of the network. Therefore, the MPC framework is highly suitable for coordination of VSLs and RM installations.

The various MPC approaches mainly differ in the prediction models, the objective functions and the control architectures. Therefore, the rest of this section outlines the prediction model, objective function and control architectures that are used in this work. For a detailed description of MPC, the reader is referred to [7].

B. Prediction Model

The macroscopic traffic model METANET [4] is used as the prediction model, as it provides a good trade-off between computational complexity and model accuracy [24] and is capable of modeling RM installations and VSLs as control inputs. It is a second-order nonlinear model that models traffic flow analogous with a compressible fluid. Consequently, the computational speed of METANET is not affected by vehicular density, making it highly suitable for model-based control.

In METANET a traffic network is modeled as a directed graph, where the links correspond to homogeneous freeway stretches with similar geometry. Typically, each link m is partitioned into segments i of length $L_{m,i}$. However, the rest of this work only differentiates between segments and does not differentiate between links to improve readability. Hence, the traffic state of each segment i at time step k is described by three aggregated variables: the traffic density $\rho_i(k)$ (in veh/km), the space-mean speed $v_i(k)$ (in km/h) and the traffic flow $q_i(k)$ (in veh/h). Moreover, the traffic state of the segments that contain an on-ramp is described by one additional variable: the vertical queue length $w_i(k)$ (in veh).

Since METANET is very well-known and frequently used in the field (see e.g., [4], [5], [17], [23]), the reader is referred to [4] for the system equations of METANET that are used in this work.

C. Objective Function

This paper uses three sub-objectives, which are added together with appropriate weights for the global network objective function.

1) Congestion Reduction: The Total Time Spent (TTS) is used as the performance criterion to minimize the congestion in the network. This is the total time that vehicles spend in a section or spend waiting at an on-ramp and is the most commonly used criterion for the reduction of congestion (see e.g., [5], [16], [25]).

2) Soft Constraints: The queue lengths at the on-ramps have to be limited to avoid spillback to urban roads. This work uses a penalty term in the objective functions that imposes soft constraints on these queue lengths. This approach, instead of hard constraints, is commonly used in literature to simplify the optimization problems and to avoid potential infeasible optimization problems (see e.g., [6], [23], [24]).

3) Signal Fluctuation: A term is included to penalize the variation in metering rates to avoid large and frequent fluctuations in the control signal. This is common practice in the field [4], [9], [23]. Undesirable fluctuations in VSLs are avoided in a different way, as will be explained in Section III-B.

Subsequently, the global objective function $\bar{J}(k_c)$, describing the network performance at control time step k_c over the prediction horizon N_p, is given by:

$$\begin{aligned}\bar{J}(k_c) = \sum_{k=Mk_c}^{M(k_c+N_p)} \Bigg[T_c \Bigg(\sum_{i\in I_{\text{all}}} \rho_i(k) L_i \lambda + \sum_{i\in I_r} w_i(k) \Bigg) \\ + \zeta_w \Bigg(\sum_{i\in I_r} \Big(\max(w_i(k) - w_{\max}, 0) \Big)^2 \Bigg) \Bigg] \\ + \sum_{j=1}^{N_p-1} \Bigg[\zeta_r \Bigg(\sum_{i\in I_r} \Big(U_r^i(k_c+j) - U_r^i(k_c+j-1) \Big)^2 \Bigg) \Bigg], \end{aligned} \tag{1}$$

where λ is the number of lanes in the network, $U_r^i(k_c) \in [0, 1]$ is the RM rate of the on-ramp at segment i at controller time

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

step k_c, M relates the control time step k_c and simulation time step k as $k = Mk_c$, T_c is the sampling time of the controller, ζ_w and ζ_r are weighting terms, w_{max} is the soft limit on the queue lengths, I_{all} is the set of all segments of the network, and I_r is the set of segments in the network that contain a controlled on-ramp.

D. Control Architectures

In Section IV, the two proposed distributed algorithms will be evaluated by comparing their performance to the more conventional centralized and decentralized algorithms.

1) Centralized MPC: With a centralized architecture, one central MPC agent determines control inputs for the entire network. The inputs are found by optimizing the global objective function using measurements of all the states and, consequently, all control measures are coordinated optimally. Hence, centralized MPC generally leads to the optimal system performance in a receding horizon context [5], [16], [22], [23]. However, it also has the largest computational complexity, as the size of the optimization problem scales with the size of the considered network.

2) Decentralized MPC: As opposed to a centralized architecture, with a decentralized architecture, the network is partitioned into subsystems and each subsystem is controlled by a local MPC agent. The agents determine local control inputs by optimizing a local objective function without communicating with each other. Consequently, decentralized MPC has the lowest computational complexity, as the size of the local optimization problems only scale with the size of the subsystems. However, generally, it also has the worst system performance in a receding horizon context [5], [16], [22], [23], as the control inputs are not coordinated at all.

3) Distributed MPC: An intermediate solution to the drawbacks of a centralized and a decentralized architecture is a distributed architecture. Similar to a decentralized architecture, the network is partitioned into subsystems and each subsystem is controlled by a local MPC agent. However, the agents are now actively communicating with each other to improve the global network performance.

Since distributed MPC is a generic term for all approaches where agents at least share some information while determining control inputs, there is a great variety in distributed MPC algorithms [26], [27]. However, with freeway traffic control, agents often share their objective function with other agents, such as the downstream neighboring agent [16], [23] or every other agent in the network [5], [23], but optimize this objective function for local control inputs. Once the agents solve their optimization problems, they communicate their intermediate solutions to the other agents, and proceed solving their optimization problems with the updated solutions of the other agents.

If the communication fails in one part of the network, the optimization problem associated with centralized MPC cannot be solved (without the estimation of the missing variables), while the local optimization problems associated with distributed MPC can still be solved (in the parts of the network without the errors). This makes distributed MPC typically more robust than centralized MPC w.r.t. communication errors.

III. Distributed MPC Using Alternating Optimization

This section proposes two novel MPC algorithms, FC-A-MPC and DC-A-MPC, for coordinated control of discrete VSLs and continuous RM rates. Both algorithms use a distributed control architecture and an alternating optimization scheme to relax the MINLP problems. The remainder of this section formalizes both algorithms.

A. Distributed Architecture

An arbitrary long freeway network is considered that is modeled by N_{all} segments indexed by $i \in I_{all}$. All segments $i \in I_r$ contain an on-ramp and all segments $i \in I_{off}$ contain an off-ramp. All on-ramps have an RM installation that can control the number of vehicles that enter the network. Furthermore, the segments $i \in I_{VSL}$ contain a matrix sign that can display speed limits. The network is subject to a mainstream demand d_0 and on-ramp demands d_i, $\forall i \in I_r$.

For the distributed architecture, the freeway is partitioned into N_{sub} subsystems, where each subsystem may contain an arbitrary number of VSLs and RM installations. Subsequently, I_{all} is partitioned into sets $\{I_1, I_2, \ldots, I_{N_{sub}}\} \subseteq I_{all}$, such that all segments in I_s are part of subsystem s. Similarly, I_r, I_{VSL}, I_{off} are partitioned into sets I_r^s, I_{VSL}^s, I_{off}^s for every subsystem s. Associated with the subsystems are N_{sub} MPC agents, which determine local control inputs for their respective subsystem.

The two proposed algorithms differ in cooperativeness. A fully cooperative architecture [5] is used with FC-A-MPC, where all agents share the same global objective function, but optimize it for local decision variables. Hence, the global objective function, based on (1) but partitioned for agent s, is formulated as:

$$\bar{J}^s(k_c) = \sum_{k=Mk_c}^{M(k_c+N_p)} \Bigg[T_c \Bigg(\sum_{i\in I_{all}} \rho_i(k) L_i \lambda + \sum_{i\in I_r} w_i(k) \Bigg) + \zeta_w \Bigg(\sum_{i\in I_r} \Big(\max(w_i(k) - w_{max}, 0) \Big)^2 \Bigg) \Bigg] + \sum_{j=1}^{N_p-1} \Bigg[\zeta_r \Bigg(\sum_{i\in I_r^s} \Big(U_r^i(k_c+j) - U_r^i(k_c+j-1) \Big)^2 \Bigg) \Bigg]. \quad (2)$$

A downstream cooperative architecture [16], [23] is used with DC-A-MPC, where every agent s optimizes a local objective function that contains the states of agent s and $s+1$. The local objective function of agent s, based on (1), is formulated as:

$$J^s(k_c) = \sum_{k=Mk_c}^{M(k_c+N_p)} \Bigg[T_c \Bigg(\sum_{i\in\{I_s, I_{s+1}\}} \rho_i(k) L_i \lambda + \sum_{i\in\{I_r^s, I_r^{s+1}\}} w_i(k) \Bigg) + \zeta_w \Bigg(\sum_{i\in\{I_r^s, I_r^{s+1}\}} \Big(\max(w_i(k) - w_{max}, 0) \Big)^2 \Bigg) \Bigg] + \sum_{j=1}^{N_p-1} \Bigg[\zeta_r \Bigg(\sum_{i\in I_r^s} \Big(U_r^i(k_c+j) - U_r^i(k_c+j-1) \Big)^2 \Bigg) \Bigg]. \quad (3)$$

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

A parallel and iterative scheme is implemented to coordinate the control inputs of the network: once all agents have solved their optimization problems, they communicate their solutions to the agents with whom they cooperate and, subsequently, proceed with solving their optimization problems with the updated solutions. This process is repeated for n_{dist} iterations.

B. Operational Constraints

As previously discussed, the fluctuations in metering rates are penalized by a term in the objective functions. Similarly, it is necessary to avoid fluctuations in the VSLs to improve driver safety and comfort [20], [28]. This is done with hard constraints because the VSLs are discrete decision variables. Hence, the hard constraints reduce the size of the solution space of the VSLs and, therefore, they reduce the complexity of the optimization problems.

Two types of constraints on the VSLs are considered. The first constraint allows the VSLs to maximally change η_{t} per controller sample:

$$\left|U_{\text{VSL}}^{i}(k_{\text{c}}) - U_{\text{VSL}}^{i}(k_{\text{c}}+1)\right| \leq \eta_{\text{t}}, \quad \forall i \in I_{\text{VSL}}, \tag{4}$$

where $U_{\text{VSL}}^{i}(k_{\text{c}})$ is the VSL at segment i at controller time step k_{c}.

The second constraint allows the VSLs that are on two consecutive freeway segments to maximally differ η_{d} from each other:

$$\left|U_{\text{VSL}}^{i}(k_{\text{c}}) - U_{\text{VSL}}^{i+1}(k_{\text{c}})\right| \leq \eta_{\text{d}}, \quad \forall i : \{i, i+1\} \subseteq I_{\text{VSL}}. \tag{5}$$

C. Alternating Optimization Scheme

The agents use an alternating optimization scheme [19] to decompose the MINLP problems into two types of subproblems: nonlinear non-convex continuous optimization problems to find RM rates, and discrete optimization problems to find VSL values. In the continuous optimization problems, the VSLs are set to be constant variables, and vice versa, in the discrete optimization problems, the RM rates are set to be constant variables. The optimization problems are solved one after another for n_{alt} iterations. In every new iteration, the constant variables are updated with the control inputs that have been found in the previous iteration.

The continuous optimization problem for agent s to find the optimal metering rates $\bar{U}_{\text{r}}^{s}(k_{\text{c}})$ in its subsystem at control time step k_{c} over the prediction horizon N_{p} is formulated as:

$$\min_{\bar{U}_{\text{r}}^{s}(k_{\text{c}})} \quad J_{\text{con}}^{s}(k_{\text{c}}) \tag{6}$$

$$\begin{aligned}
\text{subject to: } & x(k_{\text{c}}+l+1) = f_{\text{M}}\big((x(k_{\text{c}}+l), \bar{U}_{\text{r}}^{s}(k_{\text{c}}+l),\\
& \hat{U}_{\text{r}}^{s}(k_{\text{c}}+l), \hat{U}_{\text{VSL}}^{s}(k_{\text{c}}+l), d(k_{\text{c}}+l)\big),\\
& x(k_{\text{c}}) = x_{k},\\
& d(k_{\text{c}}+l) = d_{k},\\
& U_{\text{r}}^{i}(k_{\text{c}}+l) \in \mathcal{U}_{\text{r}}, \quad \forall i \in I_{\text{r}}^{s},\\
& U_{\text{r}}^{i}(k_{\text{c}}+j) = U_{\text{r}}^{i}(k_{\text{c}}+N_{c}-1), \quad \forall i \in I_{\text{r}}^{s},\\
& \text{for } j \in \{N_{\text{u}}, \ldots, N_{\text{p}}-1\},\\
& \text{for } l \in \{0, 1, \ldots, N_{\text{p}}-1\},
\end{aligned} \tag{7}$$

where J_{con}^{s} is defined in (2) for FC-A-MPC and in (3) for DC-A-MPC, $\hat{U}_{\text{r}}^{s}$ and $\hat{U}_{\text{VSL}}^{s}$ contain all the RM rates and VSLs that are constant variables in the optimization, the future states x of the network are predicted by the system dynamics f_{M} of METANET, x_{k} are the measurements of the states at time k_{c}, d are the demands in the network, d_{k} are the measurements of the demands at time k_{c}, $\mathcal{U}_{\text{r}}$ is the input space of the RM rates, and N_{u} is the control horizon that simplifies the optimization problem.

Similarly, the discrete optimization problem for agent s to find the optimal speed limits $\bar{U}_{\text{VSL}}^{s}(k_{\text{c}})$ in its subsystem at control time step k_{c} over the prediction horizon N_{p} is formulated as:

$$\min_{\bar{U}_{\text{VSL}}^{s}(k_{\text{c}})} \quad J_{\text{dis}}^{s}(k_{\text{c}}) \tag{8}$$

$$\begin{aligned}
\text{subject to: } & x(k_{\text{c}}+l+1) = f_{\text{M}}\big((x(k_{\text{c}}+l), \bar{U}_{\text{VSL}}^{s}(k_{\text{c}}+l),\\
& \hat{U}_{\text{r}}^{s}(k_{\text{c}}+l), \hat{U}_{\text{VSL}}^{s}(k_{\text{c}}+l), d(k_{\text{c}}+l)\big),\\
& x(k_{\text{c}}) = x_{k},\\
& d(k_{\text{c}}+l) = d_{k},\\
& U_{\text{VSL}}^{i}(k_{\text{c}}+l) \in \mathcal{U}_{\text{VSL}}, \quad \forall i \in I_{\text{VSL}}^{s},\\
& U_{\text{VSL}}^{i}(k_{\text{c}}+j) = U_{\text{VSL}}^{i}(k_{\text{c}}+N_{c}-1),\\
& \forall i \in I_{\text{VSL}}^{s},\\
& \text{for } j \in \{N_{\text{u}}, \ldots, N_{\text{p}}-1\},\\
& \text{for } l \in \{0, 1, \ldots, N_{\text{p}}-1\},
\end{aligned} \tag{9}$$

where $\mathcal{U}_{\text{VSL}}$ is the input space of the VSLs and J_{dis}^{s} is defined in (2) for FC-A-MPC and in (3) for DC-A-MPC.

D. Initialization and Stopping Criteria

To initialize the iterative scheme, values for the constant variables in the optimization problems of the first distributed iteration are needed. Therefore, the agents use the time-shifted VSLs of the previous controller sample for these initial values [19].

The agents use two stopping criteria to terminate the optimization scheme:

1) $timer \geq t_{\text{term}}$: the computation time of the agents, tracked by *timer*, becomes larger than t_{term}.
2) $dist = n_{\text{dist}}$: the optimization problems of the final distributed iteration n_{dist} are solved, where *dist* is the distributed iteration counter.

E. Algorithm Formulation

Both FC-A-MPC and DC-A-MPC use the same communication and optimization protocols, formulated in Algorithm 1 and illustrated in Fig. 1.

The constant variables $\hat{U}_{\text{r}}^{s}$ and $\hat{U}_{\text{VSL}}^{s}$ of agent s contain the intermediate inputs of agent s (local part) and the intermediate inputs of all the other agents with whom agent s cooperates (nonlocal part). To initialize the algorithms, the control inputs of the previous controller sample $U_{\text{VSL}}^{\text{prev}}$ and $U_{\text{r}}^{\text{prev}}$ are used as constant variables. Every agent solves the continuous optimization problem and the discrete optimization problem n_{alt} times for every distributed iteration. The local part of $\hat{U}_{\text{r}}^{s}$ is updated with $\bar{U}_{\text{r}}^{s}$ every time agent s solves the continuous optimization problem. Similarly, the local part of $\hat{U}_{\text{VSL}}^{s}$ is updated with $\bar{U}_{\text{VSL}}^{s}$ every time agent s solves the discrete optimization problem. After n_{alt} iterations, the agents communicate their optimal inputs $\bar{U}_{\text{r}}^{s}$ and $\bar{U}_{\text{VSL}}^{s}$ to the agents

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

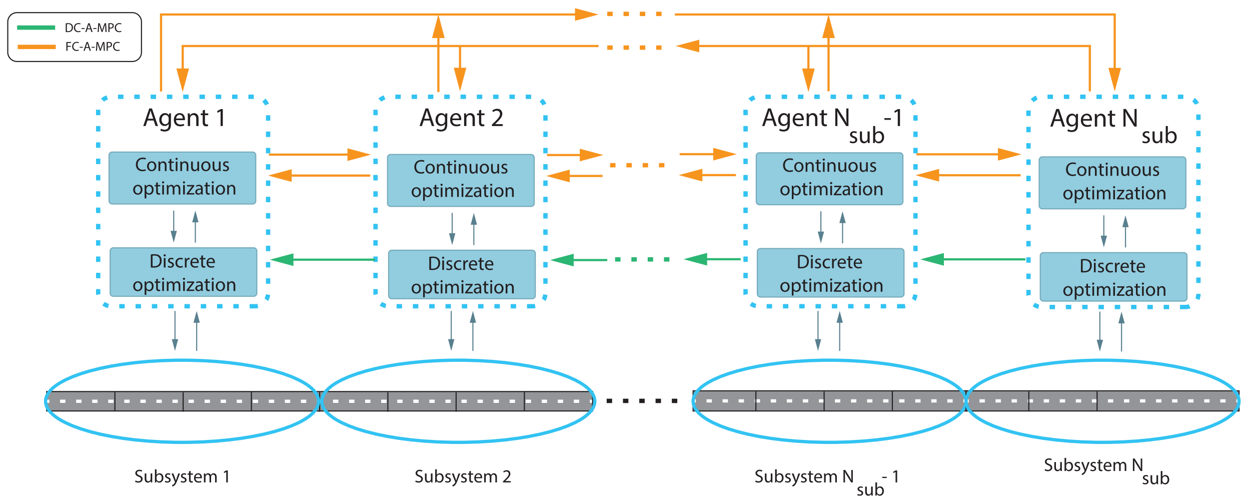


Fig. 1. A schematic representation of the communication protocols of FC-A-MPC (orange) and DC-A-MPC (green).

Algorithm 1: Top Level Communication and Optimization Protocol of FC-A-MPC and DC-A-MPC for Agent s

Input: $U_{\text{VSL}}^{\text{prev}}$, $U_{\text{r}}^{\text{prev}}$, t_{term}, n_{dist}, n_{alt}
Output: $\bar{U}_{\text{r}}^{s}$, $\bar{U}_{\text{VSL}}^{s}$
Start *timer*

Set initial values
$\hat{U}_{\text{VSL}}^{s} := U_{\text{VSL}}^{\text{prev}}$
$\hat{U}_{\text{r}}^{s} := U_{\text{r}}^{\text{prev}}$

Start iterative scheme
while *timer* $\leq t_{\text{term}}$ **do**
  for *dist = 1:* n_{dist} **do**
    for *alt = 1:* n_{alt} **do**
      Solve *continuous optimization* for $\bar{U}_{\text{r}}^{s}$
      Update local part of $\hat{U}_{\text{r}}^{s}$ with $\bar{U}_{\text{r}}^{s}$
      Solve *discrete optimization* for $\bar{U}_{\text{VSL}}^{s}$
      Update local part of $\hat{U}_{\text{VSL}}^{s}$ with $\bar{U}_{\text{VSL}}^{s}$
    end
    Store $\bar{U}_{\text{r}}^{s}$ and $\bar{U}_{\text{VSL}}^{s}$
    Communicate $\bar{U}_{\text{r}}^{s}$ and $\bar{U}_{\text{VSL}}^{s}$ to other agents
    Receive optimal inputs $\bar{U}_{\text{r}}^{\text{other}}$ and $\bar{U}_{\text{VSL}}^{\text{other}}$ of other agents
    Update nonlocal part of $\hat{U}_{\text{r}}^{s}$ with $\bar{U}_{\text{r}}^{\text{other}}$
    Update nonlocal part of $\hat{U}_{\text{VSL}}^{s}$ with $\bar{U}_{\text{VSL}}^{\text{other}}$
  end
end

with whom they cooperate and similarly receive their optimal inputs $\bar{U}_{\text{r}}^{\text{other}}$ and $\bar{U}_{\text{VSL}}^{\text{other}}$. Subsequently, all agents update the nonlocal parts of $\hat{U}_{\text{r}}^{s}$ and $\hat{U}_{\text{VSL}}^{s}$ with $\bar{U}_{\text{r}}^{\text{other}}$ and $\bar{U}_{\text{VSL}}^{\text{other}}$, and resolve to the next distributed iteration.

The algorithms are terminated either after n_{dist} distributed iterations or when the computation time is larger than t_{term}. After every distributed iteration, the local inputs that have been found are stored. Finally, the global objective function is evaluated for the inputs that have been found in every distributed iteration. The first time samples of the inputs that

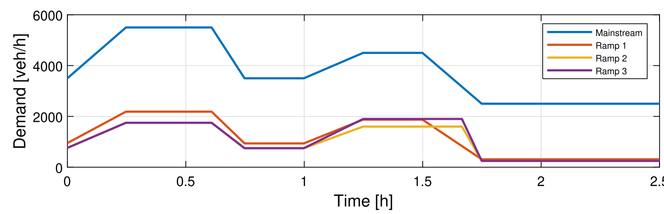


Fig. 2. Demands at the mainstream and on-ramps.

yield the best global objective function value are sent to the network as control inputs.

IV. Case Study

In this section, the proposed algorithms are evaluated for a synthetic case study. Firstly, the set-up and the traffic scenario of the case study are outlined. Secondly, performance criteria for the algorithms are defined. Then, the no-control network response is investigated. Subsequently, the control parameters of the algorithms are described. Then, the results are presented in two parts. Firstly, the influence on the system performance by relaxing the VSLs to be continuous decision variables in a distributed setting is considered. Secondly, a comparison is presented between the proposed distributed algorithms and the more conventional centralized and decentralized algorithms.

A. Case Set-Up and Scenario

The freeway network from [24] is used, but it is slightly modified to include RM installations. The network, depicted in Fig. 3, has a length of 30 km, is partitioned into $N_{\text{all}} = 24$ segments in set $I_{\text{all}} = \{1, 2, \ldots, 24\}$, contains six VSLs on the segments in the set $I_{\text{VSL}} = \{2, 3, 9, 10, 16, 17\}$, contains three RM installations at the on-ramps on the segments in $I_{\text{r}} = \{7, 14, 21\}$, and contains three off-ramps at segments in $I_{\text{off}} = \{5, 12, 19\}$. The VSLs are allowed to take values from the discrete set $\text{VSL}_{\text{set}} = \{40, 60, 80, 100\}$ km/h, as this allows a balanced trade-off between problem size and system performance. The control inputs are enumerated in the downstream direction to improve readability as illustrated in Fig. 3.

METANET, described in Section II-B, is used to simulate this network. The system parameters of the network have been chosen identically to [24]. Hence, the only model parameter

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

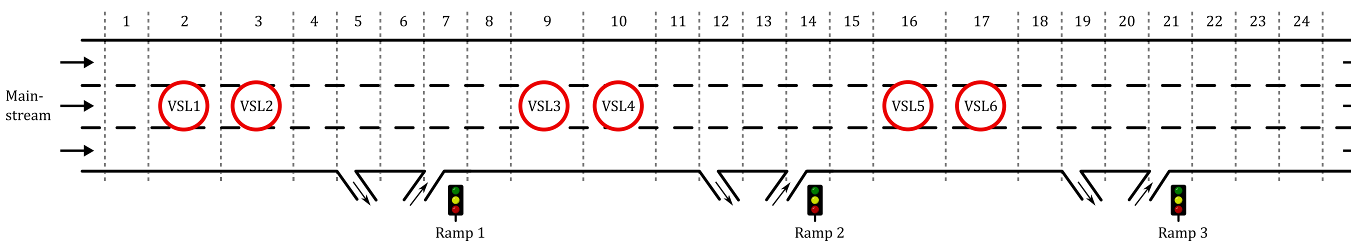


Fig. 3. An illustration of the freeway network used in the case study.

that differs per segment is the segment length. The segments that contain VSLs have a larger length, so that the speed limits have more influence on the overall traffic state of the network. To avoid spillback to hypothetical urban roads, soft constraints are imposed on the queue lengths on all three on-ramps with $w_{\max} = 100$ veh (similar values for $w_{\max}$ have been used in e.g., [4], [5], [8], [9]).

A hypothetical traffic scenario is simulated for 2.5 hours, which corresponds to $N_{\text{sim}} = 900$ samples with model sampling time $T = 10$ s. The demand profile and splitting fractions $\beta_i, \forall i \in I_{\text{off}}$ are chosen similarly to [24], but are modified so that larger traffic jams occur when no control is applied. The splitting fractions are constant during the simulations: $\beta_5(k) = 0.21$, $\beta_{12}(k) = 0.26$, $\beta_{19}(k) = 0.02$, $\forall k$. The demand profile is shown in Fig. 2.

All the simulations are conducted on an HP ZBook Studio G4, containing an Intel Core i7 processor and 8GB of RAM. The simulations are evaluated in MATLAB R2018b.

B. Performance Criteria

Two performance criteria are used to evaluate the MPC algorithms in terms of system performance and computational complexity.

1) System Performance: The TTS of the comprehensive freeway network is used to quantify the system performance:

$$\text{TTS} = \sum_{k=1}^{N_{\text{sim}}} \left[T \left(\sum_{i \in I_{\text{all}}} \rho_i(k) L_i \lambda + \sum_{i \in I_{\text{r}}} w_i(k) \right) \right]. \quad (10)$$

For convenience, this performance index is also expressed as a reduction TTS_{red} relative to the no-control case:

$$\text{TTS}_{\text{red}} = \frac{\text{TTS}_{\text{nc}} - \text{TTS}}{\text{TTS}_{\text{nc}}} \cdot 100\,\%, \quad (11)$$

where TTS_{nc} is the TTS of the no-control case.

2) Computational Complexity: To quantify the computational complexity of the algorithms, the computation time that is needed to determine the control inputs for every controller sample is investigated. For the computation time of the distributed algorithms, a summation is made of the computation times for the number of distributed iterations. The largest computation time of all controller samples is denoted as $\text{CT}_{\max}$. Hence, a control algorithm is implementable in real time if $\text{CT}_{\max}$ is smaller than the controller sampling time T_{c}.

C. No-Control System Response

In an uncontrolled setting, all ramps are open and all VSLs are equal to the maximum speed limit, such that $U_{\text{r}}^i(k_{\text{c}}) = 1, \forall i \in I_{\text{r}}, \forall k_{\text{c}}$ and $U_{\text{VSL}}^i(k_{\text{c}}) = 100$ km/h, $\forall i \in I_{\text{VSL}}, \forall k_{\text{c}}$.

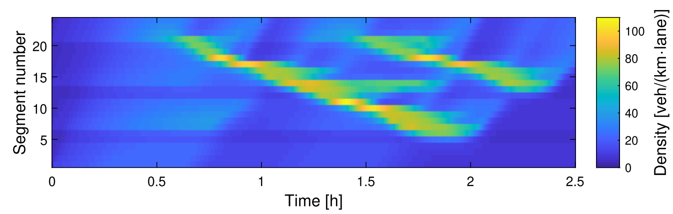


Fig. 4. Heat map of the densities in the network when no control is applied.

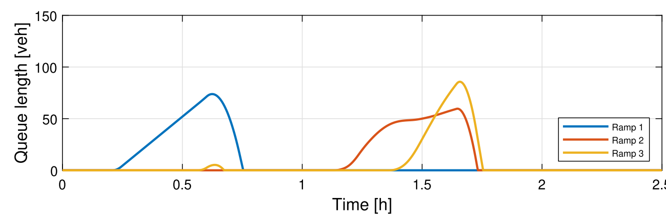


Fig. 5. Queue lengths when no control is applied.

The consequence of not controlling the high traffic demand is a major traffic congestion in the network. Two large traffic jams occur that spread over a large part of the network, as illustrated in Fig. 4. During the traffic jams, not all vehicles can freely enter the network from on-ramps. Hence, queues originate at all three on-ramps, as illustrated in Fig. 5. The no-control system performance is $\text{TTS}_{\text{nc}} = 5986$ veh·h.

D. MPC Details

The following MPC parameters are used in the case study.

1) System Partitioning: The network is partitioned into three subsystems for the decentralized and distributed algorithms. It is partitioned in such a way that each subsystems contains one RM installation and two VSLs: $I_1 = \{1, 2, \ldots, 7\}$, $I_2 = \{8, 9, \ldots, 14\}$, and $I_3 = \{15, 16, \ldots, 24\}$. Previous work has shown that the alternating optimization can be solved accurately with this combination of actuators [19].

2) Sampling Time and Horizons: All algorithms in this work use a sampling time of $T_{\text{c}} = 120$ s. To match this sampling time, the algorithms use $N_{\text{p}} = 10$. These parameters haven been chosen in such a way because the algorithms consequently take into account the network response of 1200 s in the future. This allows the algorithms, at free-flow speed, to take into account the influences of vehicles over a spatial distance of 34 km, which is slightly larger than the network size. To simplify the optimization problems, $N_{\text{u}} = 3$ is used. Hence, the horizons have been chosen in line with the rules of thumb discussed in [20].

3) Number of Alternating Iterations: The decentralized and distributed algorithms use $n_{\text{alt}} = 2$, as, in both cases, the system performance converges within two iterations.

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

The centralized algorithm requires $n_{\text{alt}} = 5$ because it has significantly larger optimization problems.

4) Weighting Terms: The weight ζ_{r} is chosen zero to avoid a tedious tuning process and to make the comparison of the different algorithms more straightforward. If less fluctuation in the RM signals is desirable, ζ_{r} can be increased.

The soft constraint weight is chosen as $\zeta_{\text{w}} = 10$, as this results in a good trade-off between queue length behavior and system performance. A 10 % violation in queue constraints is considered acceptable in this work, since a hypothetical network is used without details on the urban roads surrounding the network. If less violation is desirable, ζ_{w} can be increased.

For the operational constraints described by (4) and (5), the parameters are chosen $\eta_{\text{t}} = \eta_{\text{d}} = 20$ km/h, similarly to [19].

5) Optimization Algorithms: A comparative analysis of different solution algorithms for nonlinear freeway traffic control problems has been presented in [29]. All the continuous optimization problems in this paper are solved with Sequential Quadratic Programming (SQP) [30] combined with a multi-start approach. This approach is well known to properly solve non-convex nonlinear optimization problems and has shown good results in other works [5], [19]. The distributed and decentralized algorithms use six initial point profiles for the multi-start approach, while the centralized algorithm uses 37 initial point profiles as it has a significantly larger solution space.

The discrete optimization problems of the decentralized and distributed algorithms are solved by evaluating all the feasible solutions. Hence, this results in the global optimum of the subproblem. The size of the discrete solution space, and therefore the time needed to evaluate all the feasible solutions, increases exponentially with the size of the optimization problem. Due to this exponential growth, it is not possible to evaluate all the feasible solutions with the centralized algorithm within reasonable time. Therefore, a genetic algorithm [30] is used to solve the discrete optimization problems of the centralized algorithm.

The Optimization Toolbox and the Global Optimization Toolbox in MATLAB are used to perform the SQP and genetic algorithm optimization methods, respectively. The default SQP options of the function *fmincon* are used, where *MaxFunctionEvaluations* and *MaxIterations* are both modified to $1 \cdot 10^7$. The default genetic algorithm options of the function *ga* are used, where *PopulationSize* is modified to 800 and *MaxStallGenerations* is modified to 400.

E. Results: Relaxation on VSLs

The performance of the proposed algorithms has been compared to the performance of the Fully Cooperative Rounding Model Predictive Control (FC-R-MPC) [5] and Downstream Cooperative Rounding Model Predictive Control (DC-R-MPC) algorithms. Both these algorithms have the same distributed architectures as the proposed algorithms, but consider the VSLs as continuous decision variables and afterwards round these to acceptable values.

Two different cases have been investigated: the first case only considers one distributed iteration $n_{\text{dist}} = 1$ and the second case considers multiple distributed iterations: $n_{\text{dist}} = 4$. In both cases, the termination time is set to $t_{\text{term}} = \infty$, so that the convergence of the system performance for increasing n_{dist} can be compared.

TABLE I
THE PERFORMANCE OF DC-A-MPC AND DC-R-MPC FOR DIFFERENT NUMBERS OF DISTRIBUTED ITERATIONS

Controller	**TTS [veh·h]**	**TTS$_{\text{red}}$ [%]**	n_{dist}
DC-A-MPC	4515	24.57	1
DC-A-MPC	4505	24.74	4
DC-R-MPC	4685	21.73	1
DC-R-MPC	4612	22.95	4

TABLE II
THE PERFORMANCE OF FC-A-MPC AND FC-R-MPC FOR DIFFERENT NUMBERS OF DISTRIBUTED ITERATIONS

Controller	**TTS [veh·h]**	**TTS$_{\text{red}}$ [%]**	n_{dist}
FC-A-MPC	4481	25.14	1
FC-A-MPC	4466	25.39	4
FC-R-MPC	4679	21.83	1
FC-R-MPC	4587	23.37	4

TABLE III
THE PERFORMANCE OF CENT-A-MPC, DEC-A-MPC, FC-A-MPC, DC-A-MPC, AND THE UNCONTROLLED NETWORK

Controller	**TTS [veh·h]**	**TTS$_{\text{red}}$ [%]**	**CT$_{\text{max}}$ [s]**
No control	5986	-	0
Dec-A-MPC	4738	20.85	5
DC-A-MPC	4505	24.74	120
FC-A-MPC	4463	25.44	120
Cent-A-MPC	4452	25.63	10999

The results are summarized in Table I and II. In the worst case, the relaxation results in a performance loss of 4.42 %, corresponding to the fully cooperative architecture with one distributed iteration, where the scheme that uses alternating optimization results in $\text{TTS}_{\text{red}} = 25.14$ %, while the scheme that uses rounding results in $\text{TTS}_{\text{red}} = 21.83$ %. Moreover, the schemes that use alternating optimization find solutions much closer to the optimum with the first distributed iteration.

Hence, relaxing the VSLs to be continuous decision variables can result in a significant performance loss with a distributed architecture. This confirms the results in [19] and extends them to the distributed case.

F. Results: Architecture Comparison

The performance of the proposed distributed algorithms is now compared to the Centralized Alternating Model Predictive Control (Cent-A-MPC) and Decentralized Alternating Model Predictive Control (Dec-A-MPC) algorithms. Both algorithms also use an alternating optimization scheme, but have a centralized an decentralized architecture, respectively. The stopping criteria of the proposed algorithms is set to $t_{\text{term}} = T_{\text{c}}$ and $n_{\text{dist}} = \infty$, such that they are implementable in real time.

The performance of the four algorithms and the uncontrolled system is summarized in Table III. As expected, the centralized algorithm achieves the best system performance ($\text{TTS}_{\text{red}} = 25.63$ %). However, because only one agent controls the network by using one comprehensive system model, the computational complexity is exceptionally high. As a result, the algorithm is far from implementable in real

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

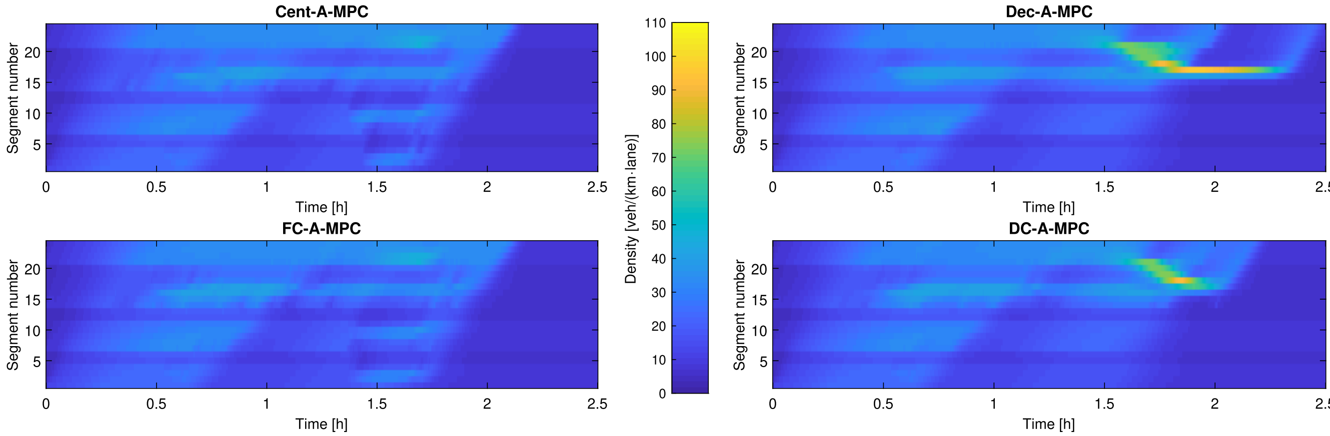


Fig. 6. Heat maps of the densities for the different algorithms.

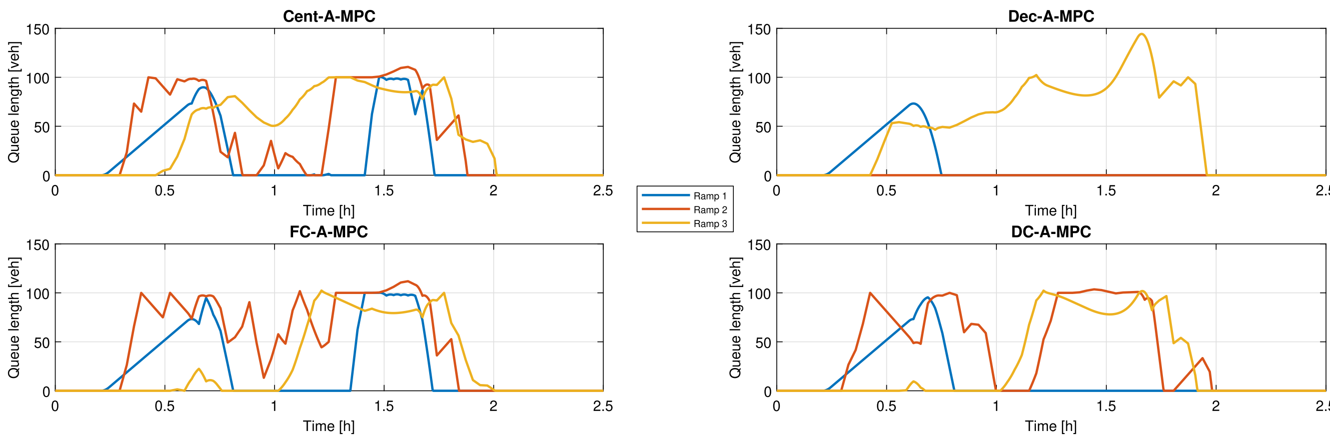


Fig. 7. Queue lengths in front of the on-ramps for the different algorithms.

time because the optimization problems cannot be solved within the controller sampling intervals. On the other hand, the decentralized algorithm has, as expected, the lowest computational complexity because three agents are controlling the network without cooperation. However, due to the lack of cooperation, the system performance is also rather sub-optimal ($\text{TTS}_{\text{red}} = 20.85\,\%$).

The proposed algorithms offer a trade-off between computational complexity and system performance. The computational complexity of the proposed algorithms is significantly lower than the computational complexity of the centralized algorithm because three agents, operating in parallel, are controlling the network. As a consequence, the proposed algorithms manage to solve the optimization problems within the controller sampling intervals. Moreover, both have a significantly better system performance than the decentralized algorithm because the agents are actively cooperating to improve the global network performance.

The system performance of FC-A-MPC ($\text{TTS}_{\text{red}} = 25.44\,\%$) is similar to the optimal performance of the centralized algorithm. Both manage to keep the network mostly uncongested, as illustrated in Fig. 6. There is a critical point in both simulations where all three on-ramps are close to the soft constraint of $w_{\max} = 100$ veh, illustrated in Fig. 7. With both algorithms, the agents decide that slightly violating the soft constraint at the second on-ramp is beneficial for the network performance. However, the queue lengths remain within 10 % of the soft constraint and the violations can be avoided if necessary by increasing ζ_{w}.

DC-A-MPC results in a worse system performance ($\text{TTS}_{\text{red}} = 24.74\,\%$) than FC-A-MPC because the agents are not fully cooperating. However, DC-A-MPC is theoretically more scalable than FC-A-MPC (therefore, less complex from a computational viewpoint), because the number of terms in the objective functions that the agents consider does not scale with size of the network as with FC-A-MPC, but with the size of the subsystems.

Both the centralized algorithm and FC-A-MPC are actively limiting vehicles at the first two on-ramps and are using the first two VSLs to avoid a traffic jam in front of the third on-ramp, as illustrated in Fig. 9. On the other hand, with DC-A-MPC, the first agent is barely limiting vehicles from entering the bottleneck at the third on-ramp, but the second agent is actively limiting the throughput to the bottleneck. As a result, a traffic jam originates in front of the third on-ramp. Lastly, with the decentralized algorithm, the first two agents are keeping their on-ramps open during the whole simulation due to the complete absence of cooperation. The result is a major traffic jam in front of the third on-ramp. Consequently, the queue at the third on-ramp violates the soft constraint

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

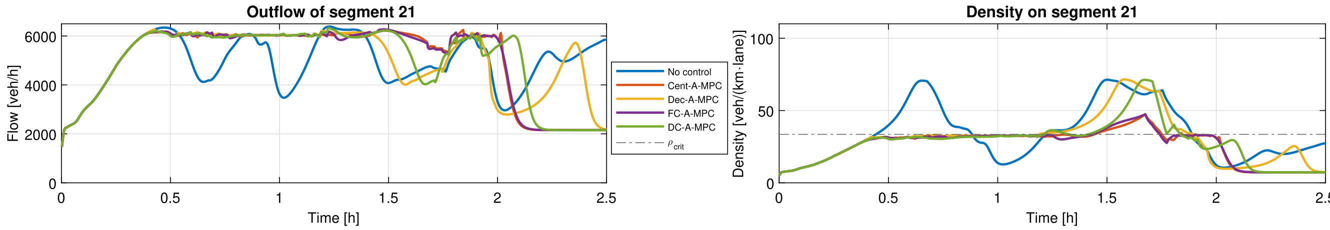


Fig. 8. The outflows of and densities on segment 21 for the different algorithms.

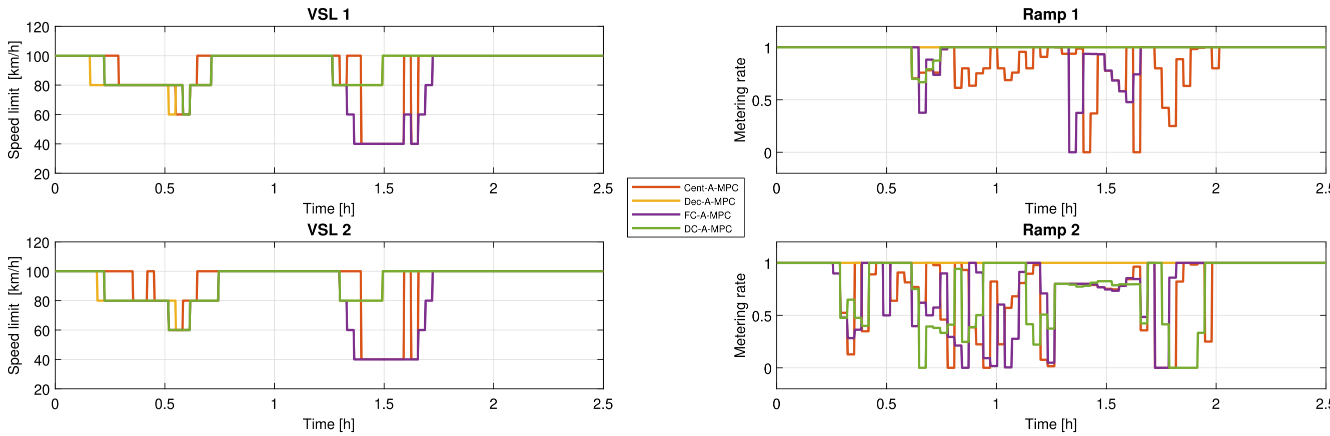


Fig. 9. The inputs for the first two VSLs and metering installations in the network for the different algorithms.

significantly. The outflows and densities of segment 21 for all algorithms are plotted in Fig. 8, as this has shown to be a bottleneck in the network due to on-ramp 3.

V. Conclusion

In this paper, two similar MPC algorithms that use a distributed control architecture and an alternating optimization scheme for coordinated control of discrete VSLs and continuous RM rates on large freeway networks have been proposed. To the best of the authors' knowledge, these are the first distributed MPC algorithms in literature that explicitly handle the VSLs as discrete decision variables in the optimization. The algorithms have been evaluated in a case study.

Firstly, the effects on the system performance by relaxing the VSLs to be continuous decision variables have been investigated for the first time with a distributed architecture. It has been found that the relaxation can result in a significant performance loss. This confirms the findings in [19] that have been found in a centralized setting, and, hence, extends those results to the distributed setting.

Secondly, the performance of the proposed algorithms has been compared to the performance of the more conventional centralized and decentralized algorithms in terms of system performance and computational complexity. Both proposed algorithms have a significantly lower computational complexity than the centralized algorithm, and, as a result, they do manage to solve all optimization problems within the controller sampling intervals. Moreover, the performance of FC-A-MPC is remarkably similar to the optimal performance of the centralized algorithm ($\text{TTS}_{\text{red}} = 25.44\,\%$ versus $\text{TTS}_{\text{red}} = 25.63\,\%$).

It depends on the size and topology of the considered freeway network which of the proposed algorithms is more suitable. For the network in the case study, FC-A-MPC has the best overall performance, as it has the best system performance of the algorithms that manage to solve all their optimization problems within the controller sampling intervals. However, unlike with FC-A-MPC, with DC-A-MPC the number of terms in the objective functions (therefore, the computational load) does not scale with the size and topology of the considered freeway network. Hence, DC-A-MPC might be more suitable for even larger freeway networks.

For future work, it would be interesting to investigate the scalability of both the proposed algorithms by evaluating their performance on even larger freeway networks. Moreover, it would also be extremely relevant to investigate the remaining challenges, mentioned in the introduction, that are to be solved in order for MPC to be implementable to large freeway networks. These include robustness analyses of MPC for freeway control, validation of the prediction models with empirical data, and the prediction of future disturbances.

References

[1] A. Hegyi and S. P. Hoogendoorn, "Dynamic speed limit control to resolve shock waves on freeways–field test results of the SPECIALIST algorithm," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 519–524.

[2] H. Hadj-Salem, J. M. Blosseville, and M. Papageorgiou, "Alinea: A local feedback control law for on-ramp metering; a real-life study," in *Proc. 3rd Int. Conf. Road Traffic Control*, 1990, pp. 194–198.

[3] M. Papageorgiou, H. Hadj-Salem, and F. Middelham, "ALINEA local ramp metering: Summary of field results," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1603, no. 1, pp. 90–98, Jan. 1997.

[4] A. Hegyi, B. De Schutter, and H. Hellendoorn, "Model predictive control for optimal coordination of ramp metering and variable speed limits," *Transp. Res. C, Emerg. Technol.*, vol. 13, no. 3, pp. 185–209, Jun. 2005.

[5] J. R. D. Frejo and E. F. Camacho, "Feasible cooperation based model predictive control for freeway traffic systems," in *Proc. IEEE Conf. Decision Control Eur. Control Conf.*, Dec. 2011, pp. 5965–5970.

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.

[6] J. R. D. Frejo and E. F. Camacho, "Global versus local MPC algorithms in freeway traffic control with ramp metering and variable speed limits," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1556–1565, Dec. 2012.

[7] J. Rawlings and D. Mayne, *Model Predictive Control: Theory Design*. San Francisco, CA, USA: Nob-Hill, 2009.

[8] J. Chen, Y. Yu, and Q. Guo, "Freeway traffic congestion reduction and environment regulation via model predictive control," *Algorithms*, vol. 12, no. 10, p. 220, Oct. 2019.

[9] P. Chanfreut, J. M. Maestre, and E. F. Camacho, "Coalitional model predictive control on freeways traffic networks," *IEEE Trans. Intell. Transp. Syst.*, early access, May 27, 2020, doi: 10.1109/TITS.2020.2994772.

[10] S. Liu *et al.*, "Robust receding horizon parameterized control for multi-class freeway networks: A tractable scenario-based approach," *Int. J. Robust Nonlinear Control*, vol. 26, no. 6, pp. 1211–1245, Apr. 2016.

[11] J. R. D. Frejo, I. Papamichail, M. Papageorgiou, and B. De Schutter, "Macroscopic modeling of variable speed limits on freeways," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 15–33, Mar. 2019.

[12] S. K. Zegeye, B. De Schuttter, J. Hellendoorn, and E. A. Breunesse, "Nonlinear MPC for the improvement of dispersion of freeway traffic emissions," in *Proc. 18th World Congr. Int. Fed. Autom. Control*, 2011, pp. 10703–10708.

[13] A. Alessandri, A. di Febbraro, A. Ferrara, and E. Punta, "Nonlinear optimization for freeway control using variable-speed signaling," *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 2042–2052, Dec. 1999.

[14] M. Papageorgiou and I. Papamichail, "Overview of traffic signal operation policies for ramp metering," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2047, no. 1, pp. 28–36, Jan. 2008.

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

[16] A. Ferrara, A. Nai Oleari, S. Sacone, and S. Siri, "Freeways as systems of systems: A distributed model predictive control scheme," *IEEE Syst. J.*, vol. 9, no. 1, pp. 312–323, Mar. 2015.

[17] N. Groot, B. De Schutter, and H. Hellendoorn, "Integrated model predictive traffic and emission control using a piecewise-affine approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 587–598, Jun. 2013.

[18] S. K. Zegeye, B. De Schutter, J. Hellendoorn, E. A. Breunesse, and A. Hegyi, "A predictive traffic controller for sustainable mobility using parameterized control policies," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1420–1429, Sep. 2012.

[19] J. R. D. Frejo, A. Nañez, B. De Schutter, and E. F. Camacho, "Hybrid model predictive control for freeway traffic using discrete speed limit signals," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 309–325, Sep. 2014.

[20] A. Hegyi, B. DeSchutter, and J. Hellendoorn, "Optimal coordination of variable speed limits to suppress shock waves," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 102–112, Mar. 2005.

[21] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Syst.*, vol. 22, no. 1, pp. 44–52, Feb. 2002.

[22] A. Ferrara, S. Sacone, and S. Siri, "Distributed model predictive control for MLD systems: Application to freeway ramp metering," in *Proc. Amer. Control Conf.*, vol. 2014, pp. 5294–5299.

[23] H. Majid, M. Hajiahmadi, B. De Schutter, H. Abouaissa, and D. Jolly, "Distributed model predictive control of freeway traffic networks: A serial partially cooperative approach," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1876–1881.

[24] J. R. D. Frejo and B. De Schutter, "SPERT: A speed limit strategy for recurrent traffic jams," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 692–703, Feb. 2019.

[25] I. M. Abuamer, M. Sadat, and C. M. J. Tampere, "A comparative evaluation of ramp metering controllers ALINEA and PI-ALINEA," in *Proc. Int. Conf. Comput. Characterization Techn. Eng. Sci. (CCTES)*, Sep. 2018, pp. 127–131.

[26] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview of features and research opportunities," in *Proc. 11th IEEE Int. Conf. Netw., Sens. Control*, Apr. 2014, pp. 530–535.

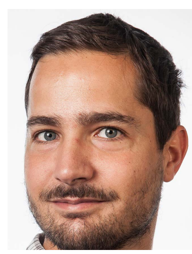
[27] R. Scattolini, "Architectures for distributed and hierarchical model predictive control – a review," *J. Process Control*, vol. 19, no. 5, pp. 723–731, May 2009.

[28] R. C. Carlson, I. Papamichail, and M. Papageorgiou, "Local feedback-based mainstream traffic flow control on motorways using variable speed limits," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1261–1276, Dec. 2011.

[29] C. Pasquale, D. Anghinolfi, S. Sacone, S. Siri, and M. Papageorgiou, "A comparative analysis of solution algorithms for nonlinear freeway traffic control problems," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1773–1778.

[30] S. S. Rao, *Engineering Optimization: Theory Practice*. Hoboken, NJ, USA: Wiley, 2009.

Uglješa Todorović received the B.Sc. degree *(cum laude)* in mechanical engineering and the M.Sc. degree in systems and control, focusing on distributed model predictive control applied to freeway networks, from the Delft University of Technology, Delft, The Netherlands, in 2016 and 2019, respectively.



José Ramón D. Frejo received the Ph.D. degree in automatic, robotic, and telematics from the University of Seville, Spain, in 2015. He was an MSCA-IF Research Fellow with the Delft University of Technology, The Netherlands, with the Delft Center for Systems and Control from 2017 to 2019. He was an Assistant Professor with the University of Seville from 2009 to 2010, in 2014, and from 2015 to 2017, and with the University Loyola Andalusia form 2014 to 2015. He was a Visiting Scholar with the University of California, Berkeley, CA, USA, in 2011, with the Delft University of Technology in 2012, and with the Technical University of Crete in 2013 and 2017. Since February 2019, he has been working as a Post-Doctoral Researcher with the ERC Advanced Grant OCONTSOLAR with the University of Seville. His research interests include model predictive control, freeway traffic control, and solar energy systems. He obtained the Outstanding Doctorate Award and the first prize of the IEEE ITS Best Dissertation Award 2016.

Bart De Schutter (Fellow, IEEE) received the Ph.D. degree *(summa cum laude)* in applied sciences from Katholieke Universiteit Leuven, Leuven, Belgium, in 1996. He is currently a Full Professor and the Head of the Department with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. His current research interests include the control of intelligent transportation systems, hybrid and discrete-event systems, and multiagent systems. He is a Senior Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Authorized licensed use limited to: TU Delft Library. Downloaded on January 23,2023 at 12:02:21 UTC from IEEE Xplore. Restrictions apply.