Structure-free model-based predictive signal control

A sensitivity analysis on a corridor with spillback

Poelman, M. C.; Hegyi, A.; Verbraeck, A.; van Lint, J. W.C.

DOI
10.1016/j.trc.2023.104174

Publication date
2023

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Transportation Research Part C 153 (2023) 104174

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Structure-free model-based predictive signal control: A sensitivity analysis on a corridor with spillback

M.C. Poelman [a,*], A. Hegyi [a], A. Verbraeck [b], J.W.C. van Lint [a]

[a] Department of Transport & Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600 GA Delft, The Netherlands

[b] Department of Multi-Actor Systems, Faculty of Technology, Policy and Management, Delft University of Technology, P.O. Box 5015, 2600 GA Delft, The Netherlands

ARTICLE INFO

Keywords:
Traffic signal control
Model predictive control
Sensitivity analysis
Prediction error

ABSTRACT

Model-based predictive signal control is a popular method to pro-actively control traffic and to reduce the effects of congestion in urban networks. In combination with structure-free controllers, which adapt signal settings in arbitrary order and combination (no imposed cycles), these predictive control methods have a high potential to increase system performance by adapting to individual vehicle patterns, which are increasingly available due to new technology. However, most of these control methods assume perfect predictions, while in practice there are prediction errors due to various reasons. In this paper, the sensitivity of the system performance to these prediction errors is analyzed, for an urban corridor with spillback.

In a microscopic simulator, first the ideal world is created for the structure-free model-based predictive signal controller, in which perfect predictions are made and the controller can reach its optimal performance. Then prediction errors are introduced in this perfect world, distinguished in aggregation errors that arise using a macroscopic prediction model and biases that represent structural errors in the prediction model or in its demand and state input. The effects of these prediction errors on the system performance are analyzed, as a function of the prediction horizon and update frequency of the control system.

The results show that, even under errors, longer prediction horizons lead to better performance, up to a certain optimal prediction horizon length. A high update frequency dampens the influence of prediction errors, enabling the structure-free controller to correct mistakes faster. However, there remains a significant performance loss due to aggregation errors and biases in the prediction model, indicating a promising performance gain of more reliable predictions and the incorporation of information on individual vehicles in future control applications. Moreover, for all model quantities one direction of the bias has more impact on the system performance than the other direction, indicating guidelines towards a more robust control system that suffers less from erroneous predictions.

* Corresponding author.
E-mail addresses: m.c.verkaik-poelman@tudelft.nl (M.C. Poelman), a.hegyi@tudelft.nl (A. Hegyi), a.verbraeck@tudelft.nl (A. Verbraeck), j.w.c.vanlint@tudelft.nl (J.W.C. van Lint).

https://doi.org/10.1016/j.trc.2023.104174
Received 5 September 2022; Received in revised form 15 February 2023; Accepted 14 May 2023
Available online 28 June 2023

1. Introduction

Traffic signal control is an important and widely used traffic management instrument to reduce the effects of congestion in urban networks, especially in saturated corridors where there is a high risk of spillback. Over the years, many signalized traffic controllers have been developed to guide traffic as efficiently as possible through the network (for an overview see (Papageorgiou et al., 2003; van Katwijk, 2008; Li et al., 2014)). Model-based predictive signal control is a popular approach, because by predicting the network wide traffic state and its evolution, the control decisions can be pro-actively optimized anticipating on future traffic conditions (for an overview see (Burger et al., 2013; Ye et al., 2019)). However, most of these predictive signal control methods assume perfect predictions, while in practice there are prediction errors due to input errors, simplifications, and biases in the prediction models. In this paper, the sensitivity to these prediction errors is investigated, analyzing the influence of prediction errors on the system performance of a model-based predictive controller in a corridor with spillback.

Model-based predictive signal control offers many possibilities to control the traffic system in an efficient manner (Burger et al., 2013; Ye et al., 2019). Since a prediction is made of the traffic state and its evolution, the controller makes pro-active decisions anticipating on future traffic conditions. The predictions are made by a traffic model, which makes it possible to calculate the effect of different possible control plans. An optimizer is used to select the control plan that is most effective according to the definition of performance in the specific control application. A model-based predictive controller uses a rolling horizon approach (as explained in (Burger et al., 2013; Ye et al., 2019)). The control plan is optimized for the upcoming prediction horizon, after which the prediction horizon is shifted, and a new control decision is made. This assures a reinitialization of the controller to the current traffic state to correct for unforeseen situations and to update the prediction regularly. By increasing the prediction horizon of the controller, information on the states of different intersections is connected. Vehicles that are released at an intersection arrive at a downstream intersection later in time, and queues at an intersection propagate backwards causing spillback at upstream intersections. Looking ahead over multiple intersections allows to optimize the traffic state for a larger part of the network (network control), instead of optimizing each individual intersection separately (local control). Therefore, in theory, the performance of the control system increases by increasing the prediction horizon, especially in saturated networks with a high chance of spillback.

To reach an adequate performance level, the controller should provide enough control plans to choose from in the optimization. Most of the existing model-based predictive control methods use a pre-defined cyclic structure, optimizing cycle times and green splits only (Burger et al., 2013; Ye et al., 2019). The more advanced predictive control systems have a larger degree of freedom, i.e., a higher adaptivity level (van Katwijk, 2008), adapt signal settings in arbitrary order and combination, and frequently (in seconds) reconsider signal settings dependent on arriving vehicles. Compared to the traditional pre-defined cyclic structure-based approaches, these structure-free predictive controllers essentially can match the fluctuations in the pattern of arriving vehicles better, improving the performance of the control system. However, these highly adaptive controllers require more computation time to find the optimal solution. Especially in the light of new technology, i.e., communicating vehicles and traffic lights, and new available data sources, i.e., online data on individual vehicles, highly adaptive controllers will be beneficial making full use of the new available data (Li et al., 2014). Therefore, structure-free (non-cyclic) model-based predictive signal controllers have a high potential controlling the traffic system in an efficient manner.

However, such control systems suffer from prediction errors when applied in real life. Often a macroscopic traffic flow model, for example the store-and-forward model (Aboudolas et al., 2009) or a related model (Lin et al., 2012), is used for computational efficiency reasons to predict the evolution of the traffic state in the network (Burger et al., 2013; Ye et al., 2019). The individual driving behavior is aggregated in the prediction model, assuming equal average driving properties (like speed, acceleration, reaction times) for all vehicles. Individual destinations are aggregated as well into average turn fractions, and average demand patterns are assumed instead of individual vehicle arrivals, since these individual vehicle data are mostly not available yet. The aggregation process leads to a model mismatch between the macroscopic prediction model and how real traffic evolves, resulting in errors in the predicted traffic states. Moreover, also biases may be present in the control system (Tettamanti et al., 2014; Tettamanti et al., 2011; Ye et al., 2017). The prediction model itself may contain biased assumptions on average driving behavior. The current traffic state, i.e., the initial state and basis of the prediction model, may be biased as well, since the traffic state, like queue length, cannot be directly measured, or is expensive to measure, and therefore needs to be estimated. The predicted arrivals and route choices, i.e., the demand input to the prediction model, can also be biased due to estimation errors. Biases in the input, initial state, and prediction model itself, lead to additional errors in the predicted traffic states. The prediction errors may propagate in the prediction model through the network, eventually leading to suboptimal control decisions and a decrease in system performance. Especially in a network with saturated traffic conditions and a high risk of spillback, small errors may have large consequences.

Existing work mainly focuses on improving data collection (Wu and Liu, 2014; Astarita et al., 2017) and estimation and prediction methods (Vlahogianni et al., 2014; van Lint and Hinsbergen, 2012). With recent technological developments, data collection shifts from historical to real-time and from location-based to floating car data (Wu and Liu, 2014; Astarita et al., 2017). Advanced estimation and prediction methods are developed using these data to improve prediction accuracy and increase computational efficiency, where not only model-based methods but also data-driven approaches are used (Vlahogianni et al., 2014; van Lint and Hinsbergen, 2012). However, until now, there has been very little study of the effect of data, estimation, and prediction errors on the performance of the signalized control system, as was already indicated for traffic management systems in general (Klunder et al., 2014), and for model-based predictive signal control systems in particular (Ye et al., 2019; Tettamanti et al., 2011; Ye et al., 2017). Whereas, when designing and implementing these predictive signalized control systems, it is important to know which prediction errors cause the most performance decrease and which improvements in prediction accuracy will lead to the largest performance increase.

Therefore, this paper studies the effect of prediction errors on the system performance of structure-free model-based predictive

signal control by means of a sensitivity analysis. This study looks at the sensitivity to aggregation errors and additional biases in the prediction model, initial state, and demand input, that all sum up to prediction errors that may influence system performance. The paper focuses on the sensitivity of structure-free model predictive control in a small network, i.e., a corridor with saturated conditions, where a small prediction error may have large consequences causing spillback. This study builds on the authors' earlier work (Poelman et al., 2020), where only a single intersection with local predictive control was analyzed. In this paper, the sensitivity analysis is extended to a corridor of multiple intersections, considering spillback effects. The dependencies between up- and downstream intersections are considered in the sensitivity analysis, analyzing the effect of propagating prediction errors in the network on the total system performance. Moreover, this paper studies the performance benefit of global predictive control with a total network objective and analyzes if this performance benefit is preserved under erroneous predictions. In the sensitivity analysis of the local controlled single intersection in (Poelman et al., 2020) only biases were considered, i.e., structural errors in the initial state (queues), input (arrivals) and output (departures) of the prediction model. However, in a network context, macroscopic prediction models are often applied to simplify the traffic propagation modeling for computational efficiency reasons. Therefore, in this paper, the prediction errors are separated into aggregation errors and additional biases in the sensitivity analysis, to identify the performance gain of more reliable predictions on the one hand and the incorporation of disaggregate individual vehicle information on the other hand. Moreover, in this study, not only errors in the initial state (queues) and input (arrivals) and output (departures) of the prediction model are analyzed, but also errors in the model itself, i.e., errors in the model parameters (travel speed, turning directions of the vehicles, saturation rate, and queue propagation speed), are considered, analyzing the model mismatch of the predicted traffic propagation through the network. Note that in authors' earlier work (Poelman et al., 2020) a comparison is made for a single intersection between structure-free and the more traditional cyclic model-based control. Since the structure-free controller outperforms the cyclic controller by adapting to individual vehicle patterns, with perfect as well as erroneous predictions (as was the conclusion in (Poelman et al., 2020)), the focus in this paper lies on the structure-free model-based predictive controller, but now applied in a network context.

The final aim of this paper is to give guidelines for developing structure-free model predictive control systems in a network. Various aspects in the sensitivity analysis are discussed, such as which prediction horizon length is still useful for looking ahead over intersections but not suffering too much from the error propagation, which update frequency is necessary to reduce prediction errors, and which quantity is most important to predict accurately in a network. Guidelines are presented towards the design of a more robust control system for a network that suffers less from erroneous predictions.

2. Methods

2.1. Experimental setup: general approach sensitivity analysis

In this research, a testbed environment is set-up to analyze the sensitivity to prediction errors for a model predictive control system. The testbed environment is outlined in Fig. 1. A microscopic simulator is used to represent the real world. A model predictive controller

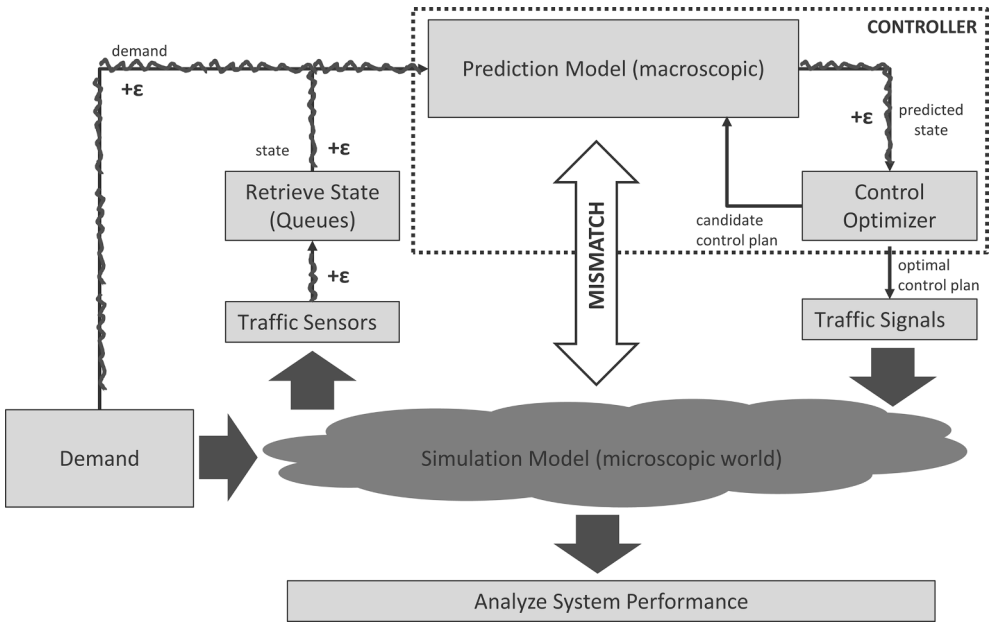


Fig. 1. Model predictive control system in testbed environment. The wavy lines represent quantities that are affected by measurement, estimation, or prediction errors ε.

is connected to the simulation platform. Based on the current traffic state and the traffic demand as defined for the simulation, the controller optimizes the settings of the traffic signals by predicting the future traffic states in the network. The controller contains the standard components of a model predictive controller (Burger et al., 2013; Ye et al., 2019). A model is used to predict the future traffic state in the network for each candidate control plan. A control optimizer is used to select the optimal control plan, that has the largest predicted system performance for the entire network. A rolling horizon approach is applied to assure a reinitialization to the current traffic state. The control plan is optimized for the upcoming prediction horizon, after which the prediction horizon is shifted, and a new control decision is made. The optimal control plan is sent to the microscopic simulator, where the effects of the control decision are simulated. Afterwards, the performance of the control system is analyzed based on the simulation results.

In contrast to real-life, all demand and state inputs to the model predictive controller are known exactly from the simulator in the testbed environment. Moreover, the prediction model can be tuned to reproduce the known and neat traffic behavior in the microscopic simulator as close as possible. Therefore, in the testbed environment, the controller can make the most accurate predictions (only a small tuning error remains), leading to an optimal control decision with optimal system performance. This provides us a clear reference situation for the sensitivity analysis. Prediction errors that arise in real-life can be systematically introduced in the testbed environment. These prediction errors will eventually affect the solution obtained by the controller, resulting in non-optimal decisions. The effect of these non-optimal decisions on the system performance can be exactly measured in the microscopic simulation world. In the sensitivity analysis the effect of these prediction errors is analyzed and compared to the reference situation. Note that a macroscopic prediction model is chosen, which is mostly applied in real-life control applications for performance reasons. This allows us to study the effects of model mismatch between the macroscopic prediction and microscopic world that would occur in real life.

The general approach of the sensitivity analysis is outlined in Fig. 2. The sensitivity analysis consists of three steps. First, the almost perfect world is created to use as a reference situation (see step I in Fig. 2). The almost perfect world represents the ideal world without prediction errors. In this almost perfect world, the predictions are made as accurate as possible. To this end, deterministic driving behavior is considered in the microscopic simulation, assuming all vehicles having equal driving parameters (like accelerations, decelerations, desired speeds, and reaction times). Drivers do not overtake and only change lanes at link exits if this is needed to reach their destination. The macroscopic prediction model is tuned to match the deterministic microscopic behavior, by measuring the values of the macroscopic parameters (like travel times, saturation rates, and queue propagation speeds) in the microscopic simulation. Moreover, individual vehicle information on arrivals and destinations is retrieved from the microscopic model and is assumed to be known in the macroscopic model as the specific demand pattern (including the turning directions of the vehicles). Also, the current

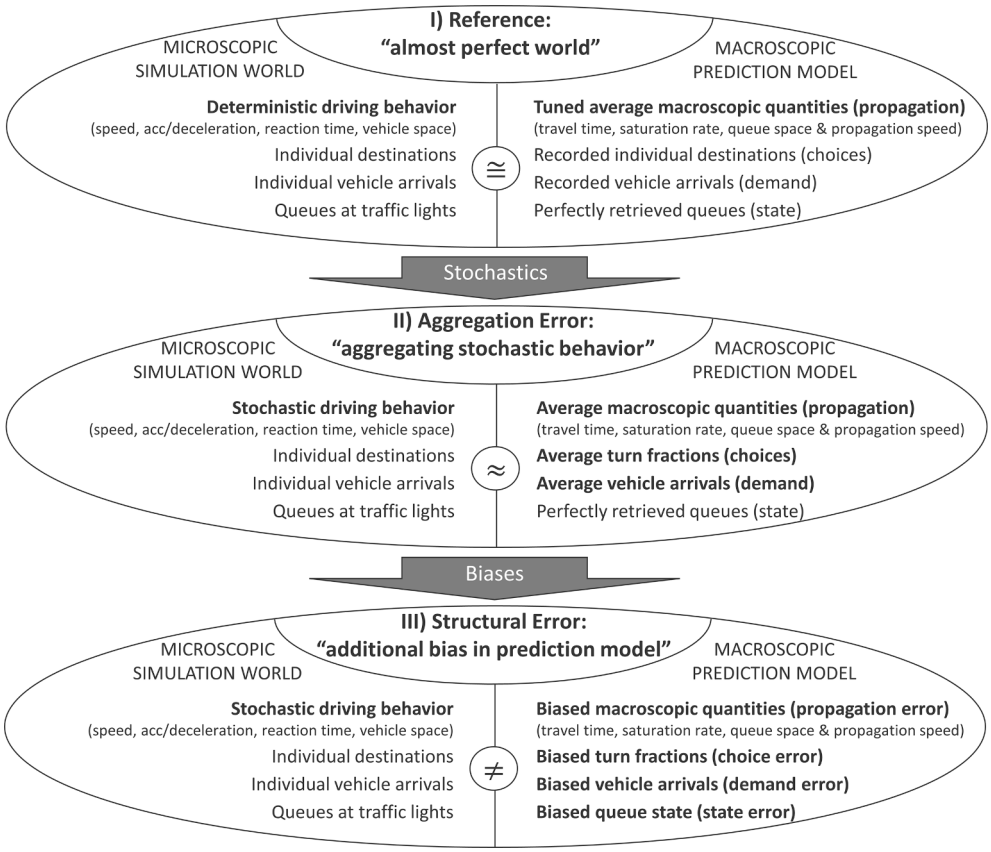


Fig. 2. General approach of the sensitivity analysis.

queue state is retrieved from the microscopic simulator and assumed to be known exactly as a starting point for the macroscopic prediction. In this way, the macroscopic model can reproduce the microscopic world almost perfectly with its predictions (only a small tuning error remains), and the model predictive controller can find the optimal control decisions leading to the optimal system performance.

Second, the almost perfect world is disturbed by introducing stochastics (see step II in Fig. 2). The deterministic driving behavior in the microscopic simulation is replaced by stochastic behavior. Vehicles are assumed to have different driving parameters, all varying around the original average values. Note that only longitudinal driving behavior is considered, lane change behavior due to overtaking is not considered in this study. The tuned macroscopic model no longer represents the exact behavior of all vehicles, but now corresponds to the average driving behavior. Moreover, the individual arrivals and destinations of the vehicles are assumed to be unknown in the macroscopic model and are replaced by the average demand pattern (including average turn fractions to indicate the distribution over the different directions). In fact, the stochastic behavior of the individual vehicles is aggregated in the macroscopic prediction model, resulting in aggregation errors. This represents the aggregation process in real life, that is inevitable when using a macroscopic model to predict microscopic behavior. The effect of these aggregation errors on the performance of the model predictive control system is analyzed.

Third, the aggregated world is disturbed further by introducing biases, i.e., structural asymmetric errors (see step III in Fig. 2). These biases are introduced in the different components of the model predictive control system (as indicated by errors ε in Fig. 1). The macroscopic prediction model itself is biased, by adding biases to the macroscopic model parameters (like travel times, saturation rates, and queue propagation speeds), representing a mismatch in driving behavior between the macroscopic prediction model and the microscopic world. The average demand input (including turn fractions) to the macroscopic prediction model is biased as well, representing demand prediction errors (and lack of information on vehicle destinations). Moreover, the initial queue state is biased, representing a combination of detection and state estimation errors that appear in real-life. The effect of these biases on the system performance is analyzed.

Note that the approach of the sensitivity analysis is top down, disturbances are introduced in the almost perfect world and the (negative) effects of these prediction errors on the performance of the control system are analyzed. Bottom up the flow chart can be considered as the improvement of the control system, from the current imperfect control system based on biased macroscopic predictions, to an 'ideal' control system with perfect predictions including information on individual vehicles. Instead of analyzing the loss in performance by prediction errors, the gain in performance can be indicated of these methodological and technological improvements of the control system.

In the sensitivity analysis in this paper, the focus lies on the network effects of prediction errors. Prediction errors propagate through the network in downstream as well as in upstream directions. Errors in predicted departures at an intersection will result in errors in the predicted arrivals at downstream intersections later in time. Errors in predicted queue states will affect the predicted departures at upstream intersections due to spillback. The longer the prediction horizon, the more the predicted state may deviate from the actual state to a point beyond which prediction may not be beneficial anymore. Therefore, the role of the prediction horizon will be considered in the sensitivity analysis. The prediction horizon is increased to look over multiple intersections, as well in downstream direction (travel direction of the vehicles) as in upstream direction (spillback direction), and the effect on the system performance is analyzed for perfect as well as erroneous conditions. For perfect predictions, it is analyzed how far the controller should look ahead to substantially increase performance, as there may be a horizon beyond which system performance does not improve anymore. For erroneous predictions, it is analyzed when the performance loss due to the error propagation overrides the benefit of the prediction. Note that updating the prediction and control decision more frequently based on (sufficiently accurate) actual queue states will probably reduce the effect of the error propagation. Therefore, the role of the update frequency will be considered in the sensitivity analysis as well. More specifically, the following aspects will be assessed:

- Relation between the prediction horizon and the performance of a control system with perfect predictions, to analyze the performance gain of an increasing prediction horizon in a network. And, to find the horizon (if any) beyond which system performance does not improve anymore.
- Relation between the prediction horizon and the performance of a control system with aggregation errors in the predictions, to analyze to what extent an increasing prediction horizon still increases system performance in a network. In other words, to analyze whether a macroscopic model is accurate enough to benefit from the prediction in the controller.
- Relation between the prediction horizon and the performance of a control system with biases in the predictions, to analyze to what extent an increasing prediction horizon increases the system performance in a network. Or, if not, to analyze to what extent the performance loss due to the error propagation in the network overrides the benefit of the prediction.
- Ranking of quantities to which the control system is most sensitive if not predicted correctly. In other words, ranking of quantities in order of importance to make accurate predictions in a network.
- Relation between the update frequency and the performance of a control system with predictions errors (aggregation errors or biases), to analyze to what extent a higher update frequency has a reducing effect on the performance loss due to error propagation in the network.

2.2. Technical setup: model predictive control system

2.2.1. Modeling framework

The modeling framework for the sensitivity analysis is an extension of the framework proposed in the authors' earlier work for a

single intersection (Poelman et al., 2020), now being applied to a small network. A microscopic simulation model in Aimsun (Aimsun, 2017) is used to represent the real world. A model predictive controller is implemented and connected to the Aimsun simulation model. The controller is assumed to be structure-free, i.e., no cycles are imposed and the order in which the traffic signals become green together with the green duration is determined in the optimization process. The authors have previously shown (Poelman et al., 2020) that for a single intersection structure-free control outperforms model-based cyclic control even under erroneous conditions due to its high adaptive ability to correct bad decisions quickly. The prediction model in the structure-free controller is an extension of the macroscopic store-and-forward model of (Poelman et al., 2020) including simple queue dynamics to predict spillback effects between the different intersections in the network. Note that the macroscopic network model is defined in enough detail containing the essential quantities to describe the microscopic traffic propagation but simplified enough to maintain a reasonable computational performance. The essential macroscopic model quantities are tuned to reproduce the reference and detuned in the sensitivity analysis (Section 2.2.4). Moreover, the predictive controller is a network controller, i.e., it optimizes the future traffic situation in the entire network, making a control decision for all intersections simultaneously. Since a rolling horizon approach is applied, a series of subsequent optimization problems is solved. In each optimization problem, the control optimizer determines the optimal control sequence for the prediction horizon that minimizes the total delay in the network based on the predicted traffic states. Each optimization problem can be defined as a discrete mathematical programming problem (Section 2.2.2). Note that the problem complexity of the structure-free model-based predictive controller rapidly grows with additional intersections, therefore, the exact solution method of (Poelman et al., 2020) is no longer sufficient and a heuristic solution method is needed and proposed in this paper (Section 2.2.3).

2.2.2. Mathematical formulation

To this end, the continuous time horizon is split into discrete control intervals $[(k\text{-}1)T, kT)$, with k the discrete time index referring to the kth interval and T [s] the duration of the time interval. At the start of each control interval the signal setting at the intersections may change. Let i denote the index of a movement at any of the intersections in the network. Let m denote the index of an intersection. Predefine, per single intersection, a movement group with index j_m as a group of movements belonging to intersection m that are non-conflicting and are given green at the same time. Let $I(j_m)$ be the set of movement indices belonging to movement group j_m. Introduce signal states $s_i(k) \in \{0(\text{red}), 1(\text{green})\}$ and movement group states $p_{j_m}(k) \in \{0(\text{red}), 1(\text{green})\}$ for all movements i, movement groups j_m at intersections m, and time indices k. Exactly one movement group can be active at each intersection m at each time index k, i.e., $\sum_{j_m} p_{j_m}(k) = 1$. The states of the signals $i \in I(j_m)$, belonging to movement group j_m at intersection m, follow the state of this movement group at each time index k, i.e., $s_i(k) = p_{j_m}(k)$. There is no all-red movement group defined. Individual movements that are switching from state red to green, are assumed to stay red a loss-time T_L longer in the next control interval (assuming $T_L < T$) representing clearance times. There is a free choice of the active movement group at an intersection, there is no pre-defined order and there are no additional constraints concerning minimum or maximum green times. The objective of the controller is to find a control sequence of movement group states $p_{j_m}(k) \in \{0,1\} \forall j_m \forall m$, and corresponding signal states $s_i(k) \in \{0,1\} \forall i$, for the prediction horizon $k = k_0 + 1, \ldots, k_0 + K$ that minimize the total delay over this prediction horizon, with k_0 referring to the current time interval the traffic system is in and K the number of prediction intervals looked ahead.

To predict the delay corresponding to a control sequence, a macroscopic store-and-forward model is used. In this model, vertical queues are assumed at the downstream end of each dedicated link of a movement. Vehicles are assumed to travel with equal free flow speed to the downstream end of the link and enter the queue. Vehicles leave the queue when the signal is green, and if there is enough space at the dedicated links of the downstream movements. To determine the available space downstream the queues are assumed to be horizontal, and a simple queueing model is applied to determine the occupied space (head and tail) of the queue. According to these assumptions, a vehicle is queued and delayed if it arrived at the downstream end of the link according to free flow conditions but has not departed yet due to a red signal, or queued earlier arrived vehicles, or a blockage downstream. The number of queued (and delayed) vehicles $x_i(k)$ [veh] per movement i is defined as:

$$x_i(k) = x_i(k-1) + a_i(k) - d_i(k) \;\; \forall i \;\forall k \tag{1}$$

with arrivals $a_i(k)$ [veh] at the queue and departures $d_i(k)$ [veh] from the queue, and $x_i(k_0)$ [veh] initialized to the current system state.

The arrivals at the queues of the external movements at the entrance of the network are equal to the demand. The arrivals at the queues of the internal movements are related to the departures of the queues of the upstream movements by:

$$a_i(k) = \sum_{\tilde{i}} (d_{\tilde{i}}(k - n_i) * \pi_{\tilde{i}\,i}) \;\; \forall i \;\forall k \tag{2}$$

where $\pi_{\tilde{i}\,i} \in [0,1]$ is the turn fraction of the vehicles leaving at upstream movement $\tilde{i}$ heading for movement i, and n_i is the number of intervals looking back upstream according to the free flow travel time, i.e.,

$$n_i = \lceil (L_i / v_i)/T \rceil \quad \forall i \tag{3}$$

with L_i [m] the length and v_i [m/s] the free flow speed of the dedicated link of movement i.

The departures from the queue are first determined by calculating the desired departures, considering unlimited space downstream, and, if necessary, will be restricted afterwards. The desired departures $\bar{d}_i(k)$ [veh] from the queue of movement i, are approximated by an average saturation flow rate r_i [veh/s] multiplied by the time duration T (corrected with the loss time $T_L \leq T$ when switching from red to green), i.e.,

$$\bar{d}_i(k) = \begin{cases} 0 & \text{if } s_i(k) = 0 \\ \min\{r_i{*}(T - T_L), x_i(k-1) + a_i(k)\} & \text{if } s_i(k) = 1 \text{ and } s_i(k-1) = 0 \\ \min\{r_i{*}T, x_i(k-1) + a_i(k)\} & \text{if } s_i(k) = 1 \text{ and } s_i(k-1) = 1 \end{cases} \quad \forall i \ \forall k \tag{4}$$

These desired departures $\bar{d}_i(k)$ [veh] from movement i, are limited by a maximum outflow $d_i^{\text{MAX}}(k)$ due to limited space at the downstream movements $\hat{\imath}$, resulting in the real departures $d_i(k)$ [veh], i.e.,

$$d_i(k) = \min\left\{\bar{d}_i(k), d_i^{\text{MAX}}(k)\right\} \forall i \ \forall k \tag{5}$$

with

$$d_i^{\text{MAX}}(k) = \min_{\hat{\imath}}\left\{\frac{\left(L_{\hat{\imath}} - L_{\hat{\imath}}^Q(k-1)\right)/L_{\text{VEH}} - \sum_{n=1}^{n_i}\sum_{\hat{\imath}}(d_{\hat{\imath}}(k-n){*}\pi_{\hat{\imath}\,\hat{\imath}})}{\pi_{i\,\hat{\imath}}}\right\} \forall i \ \forall k \tag{6}$$

where $\pi_{i\,\hat{\imath}} \in [0,1]$ is the turn fraction of the vehicles leaving movement i, heading for downstream movement $\hat{\imath}$, and $\left(L_{\hat{\imath}} - L_{\hat{\imath}}^Q(k-1)\right)/L_{\text{VEH}}$ the available space expressed in vehicles at the downstream link of movement $\hat{\imath}$, i.e., link length $L_{\hat{\imath}}$ [m] minus the space occupied by the queue $L_{\hat{\imath}}^Q(k-1)$[m] divided by the space of a single queued vehicle L_{VEH} [m], and $\sum_{n=1}^{n_i}\sum_{\hat{\imath}}(d_{\hat{\imath}}(k-n){*}\pi_{\hat{\imath}\,\hat{\imath}})$ the number of driving vehicles already present on downstream movement $\hat{\imath}$. Note that $d_i(k)$ and $a_i(k)$ only depend on values of previous time indices, so the order in which the links of the different movements i are updated has no influence on the calculation.

The space occupied by the queue $L_i^Q(k)$ [m] in (6) is defined by the tail position of the queue. A simple (first order) queuing model is used to keep track of the tail and head position of the queue. The tail $L_i^Q(k)$ and head $L_i^H(k)$ of the queue are expressed in the positive number of meters in upstream direction measured from the downstream end of the link. The tail of the queue moves upstream with the arrivals, and remains active until the head of the queue, propagating upstream with constant speed v_i^H [m/s] $< v_i$ when releasing vehicles, meets the tail position, i.e.,

$$L_i^Q(k) = \begin{cases} L_i^Q(k-1) + a_i(k){*}L_{\text{VEH}} & \text{if } L_i^H(k-1) < L_i^Q(k-1) \text{and} (s_i(k) = 0 \text{ or } s_i(k-1) = 1) \\ x_i(k){*}L_{\text{VEH}} & \text{otherwise ((re)initialization)} \end{cases} \forall i \ \forall k \tag{7}$$

and

$$L_i^H(k) = \begin{cases} L_i^H(k-1) + v_i^H{*}T & \text{if } 0 < L_i^H(k-1) < L_i^Q(k-1) \text{and} (s_i(k) = 0 \text{ or } s_i(k-1) = 1) \\ v_i^H{*}(T - T_L){*}1_{[d_i(k)>0]}{*}1_{[x_i(k)>0]} & \text{otherwise ((re)initialization)} \end{cases} \forall i \ \forall k \tag{8}$$

with $1_{[d_i(k)>0]}$ and $1_{[x_i(k)>0]}$ indicator functions, 1 if the condition holds, 0 otherwise. In the model a standing queue at the stop line is assumed the moment the signal turns green, i.e., $s_i(k) = 1$ and $s_i(k-1) = 0$. The tail $L_i^Q(k)$ of the queue is initialized to the (occupied space of the) total number of vehicles in this queue (initialization in (7)). From there the tail moves upstream with the arrivals. The head $L_i^H(k)$ of the queue is initialized to zero and starts to move when releasing vehicles at green, i.e., if $d_i(k) > 0$ and $x_i(k) > 0$ (initialization in (8)). The head continuous to move upstream with constant speed v_i^H while the head is smaller than the tail, i.e., $L_i^H(k-1) < L_i^Q(k-1)$. If the head meets the tail, the head and tail are reinitialized. If the queue is dissolved during green ($x_i(k) = 0$), then the head and tail will both be reset to zero. If there is a remaining queue ($x_i(k) > 0$) since the traffic signal turns red before the head meets the tail, the head will be reset to zero ($d_i(k) = 0$) and the tail will be reset to the (occupied space of the) total number of vehicles left in the queue. Note that the first order model is only accurate if the head meets the tail before the signal gets green again for a second time. Note that (7) and (8) force a reinitialization if this is not the case. In this paper only short distances between intersections will be considered, resulting in short queues for which this queueing model is accurate enough.

According to the complete store-and-forward model, the optimization of delay of the predictive control system can be expressed as:

$$\min_{\{p_{j_m}(k), \forall j_m \forall m \forall k\}} \sum_i \sum_{k=k_0+1}^{k_0+K} x_i(k){*}T \tag{9}$$

s.t.

$$\begin{aligned} & s_i(k) = p_{j_m}(k) && \forall i \in I(j_m) \ \forall j_m \forall m \ \forall k \\ & \textstyle\sum_{j_m} p_{j_m}(k) = 1 && \forall m \ \forall k \\ & x_i(k) \text{given by (1)-(8)} && \forall i \ \forall k \end{aligned}$$

2.2.3. Solution method

The discrete mathematical programming problem (9) with the explicit formulation of the store-and-forward model (1)-(8) can be solved by searching through a corresponding decision tree. The starting node of the tree is the current state of the control system $\{x_i(k_0) \ \forall i\}$, branches are formed by the possible decisions for the next time step $k_0 + 1$, i.e., for all intersections m the choice of the active movement group j_m for which $p_{j_m}(k_0 + 1) = 1$, resulting in nodes at the next time layer with state $\{x_i(k_0 + 1) \ \forall i\}$, and so on, adding time layers for $k = k_0 + 1, \ldots, k_0 + K$. The optimal decision sequence is the path in the tree with minimum delay. The problem size, given by the number of possible paths $(|J|^{\wedge}|M|)^{\wedge}K$, rapidly grows with the width of the tree, determined by the combination $(|J|^{\wedge}|M|)$ of the

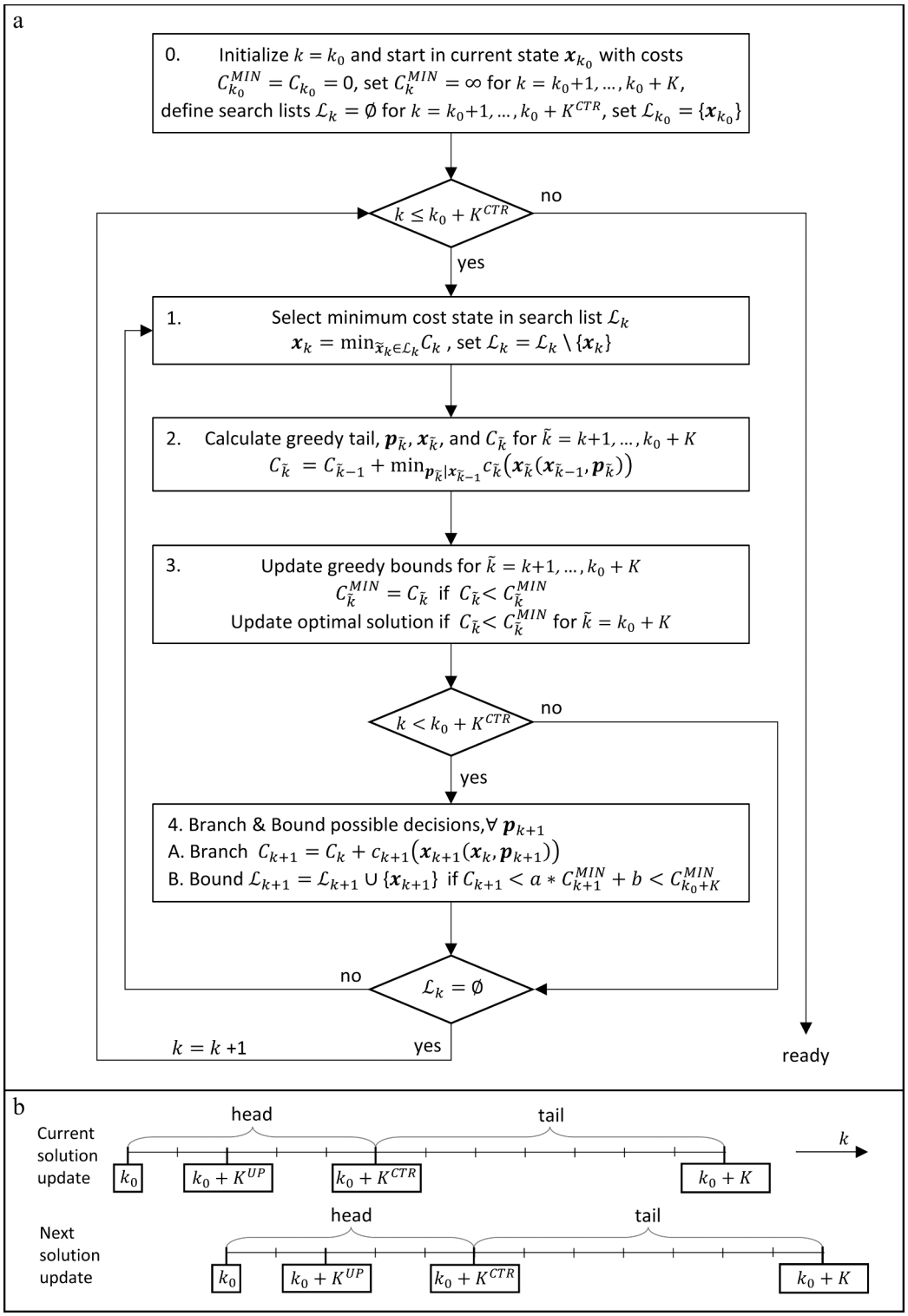


Fig. 3. Branch-and-bound heuristic with greedy tail, with flowchart (a,b) of the heuristic solution algorithm, and convergence (c,d) of the heuristic solution approach indexed to the optimal solution level of the full branch-and-bound search method (timed on a computer with an Intel i7-6820HQ processor).

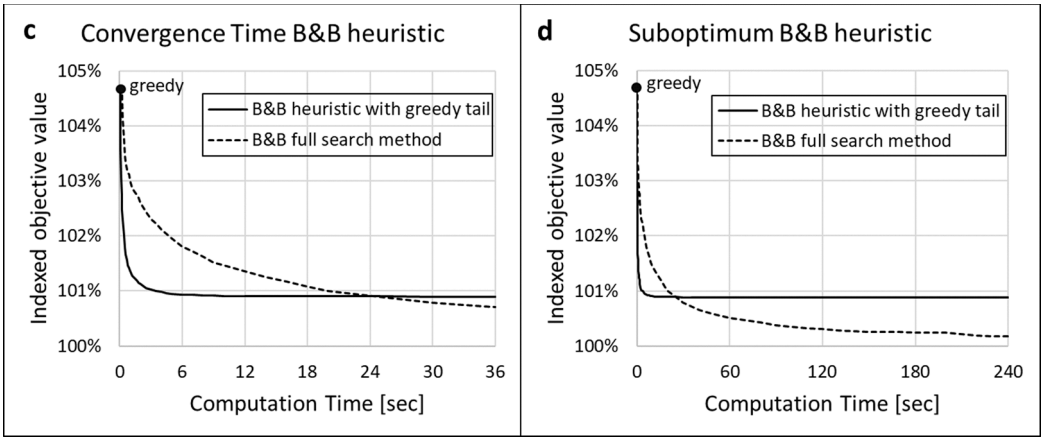


Fig. 3. (*continued*).

average number of movement groups $|J|$ per intersection and the number of intersections $|M|$, together with the depth of the tree, given by the number of prediction intervals K. Since the problem size can already be large in a small network, heuristic approaches are needed in the search process, to find a near optimal solution in acceptable time.

The applied heuristic solution approach is outlined in Fig. 3b. The rolling prediction horizon $k = k_0 + 1, \ldots, k_0 + K$ is split into a control horizon $k = k_0 + 1, \ldots, k_0 + K^{CTR}$, $K^{CTR} \leq K$, where the signal states are fully optimized and (if $K^{CTR} < K$) a tail $k = k_0 + K^{CTR} + 1, \ldots, k_0 + K$, where the search space is limited, and suboptimal signal states are obtained (Fig. 3b). Note that in the rolling horizon approach, the solution is updated after a pre-defined period of K^{UP} intervals. The control horizon $k = k_0 + 1, \ldots, k_0 + K^{CTR}$ is chosen at least as large as this update horizon $k = k_0 + 1, \ldots, k_0 + K^{UP}$, $K^{UP} \leq K^{CTR} \leq K$. In this way, only the fully optimized signal states are implemented, and the suboptimal tail is reconsidered and optimized further in the next solution update. In this paper, for performance reasons K^{CTR} is chosen as small as possible, i.e., $K^{CTR} = K^{UP}$, although not smaller than 2 intervals to assure enough degrees of freedom in the decision sequence.

The optimal solution in the tail of the decision sequence is approximated by the greedy solution, i.e., in each time step the control decision is made that provides the most performance gain in this time step, without considering the effects of this decision for the remaining horizon. The greedy solution can be calculated efficiently but is typically a suboptimal solution. The head of the decision sequence for the control horizon is fully optimized by a branch-and-bound approach. This full optimization is more costly, but an exact solution can be obtained. Note that the greedy tail itself is not part of the implemented solution and will be used as a starting point to optimize further in the next decision update. The greedy tail is determined by solving subsequent optimization problems, i.e., $\min_{\{p_{jm}(k), \forall j_m \forall m\}} \sum_i x_i(k) * T$ given state $\{x_i(k\text{-}1)\ \forall i\}$ sequentially for $k = k_0 + K^{CTR} + 1, \ldots, k_0 + K$, and is in fact constructed by following the greedy branch from a given node in layer k-1 to the next layer k in the tree. Mathematically, the objective of the controller in (9), is approximated by:

$$\min_{\{\boldsymbol{p}_k, k=k_0+1, \cdots, k_0+K^{\mathrm{CTR}}\}} \left(\sum_{k=k_0+1}^{k_0+K^{\mathrm{CTR}}} c_k(\boldsymbol{x}_k(\boldsymbol{x}_{k-1}, \boldsymbol{p}_k)) + \sum_{\tilde{k}=k_0+K^{\mathrm{CTR}}+1}^{k_0+K} \min_{\boldsymbol{p}_{\tilde{k}} | \boldsymbol{x}_{\tilde{k}-1}} c_{\tilde{k}}\left(\boldsymbol{x}_{\tilde{k}}\left(\boldsymbol{x}_{\tilde{k}-1}, \boldsymbol{p}_{\tilde{k}}\right)\right) \right) \tag{10}$$

with $\boldsymbol{x}_k$ the vector of queue states $x_i(k)\ \forall i$, $\boldsymbol{p}_k$ the vector of movement group states $p_{j_m}(k), \forall j_m \forall m$, and c_k the cost for time index k as a function of the queue state vector $\boldsymbol{x}_k$, i.e., $c_k(\boldsymbol{x}_k) = \sum_i x_i(k) * T$. Note that $\boldsymbol{x}_k$ depends on $\boldsymbol{x}_{k-1}$ and $\boldsymbol{p}_k$ through (1)-(8), that are defined in explicit form. Note that the costs are additive over time, and can be updated cumulatively, $C_k = C_{k-1} + c_k$, with C_k the cumulative costs for $k = k_0 + 1, \ldots, k_0 + K$ and $C_{k_0} = 0$. Therefore a branch-and-bound tree search can be followed to find the solution to (10). Note that the greedy solution can be used as an initial solution to (10) for the complete horizon, setting the initial cost bound in the search process, i.e., $C_{k_0+K} = \sum_{k=k_0+1}^{k_0+K} \min_{\boldsymbol{p}_{\tilde{k}} | \boldsymbol{x}_{\tilde{k}-1}} c_{\tilde{k}}\left(\boldsymbol{x}_{\tilde{k}}\left(\boldsymbol{x}_{\tilde{k}-1}, \boldsymbol{p}_{\tilde{k}}\right)\right)$.

The branch-and-bound algorithm with greedy tail is presented in Fig. 3a. The search process starts at the first layer of the tree, i.e., at time index k_0 (Step 0). For the active time layer k, the node with the lowest costs C_k is selected. Note that for time layer k_0 the start node of the tree is selected, i.e., current state $\boldsymbol{x}_{k_0}$ with $C_{k_0} = 0$ (Step 1). First the greedy path from the active node to the end of the tree is determined (Step 2). The costs along the greedy path, $C_{\tilde{k}}$ for $\tilde{k} = k + 1, \ldots, k_0 + K$ are used to determine the minimum cumulative costs $C_{\tilde{k}}^{\mathrm{MIN}}$ found so far at each layer. These optimal layer costs are used later to limit the search space. Note that for time layer $\tilde{k} = k_0 + K$ the minimum costs $C_{\tilde{k}}^{\mathrm{MIN}}$ corresponds to the best complete solution found so far, that is updated if $C_{\tilde{k}} < C_{\tilde{k}}^{\mathrm{MIN}}$ (Step 3). Then the active node is branched to the next time layer $k + 1$, by considering all possible decisions $\boldsymbol{p}_{k+1}$, calculating new states $\boldsymbol{x}_{k+1}$ with costs c_{k+1} and updating cumulative costs $C_{k+1} = C_k + c_{k+1}$ (Step 4A). The branched nodes are bounded if the costs are larger than the complete solution costs found so far, i.e., $C_{k+1} \geq C_{k_0+K}^{\mathrm{MIN}}$ or if the costs are too large compared to the minimum cumulative layer costs

C^{MIN}_{k+1}, i.e., $C_{k+1} \geq a^{*}C^{MIN}_{k+1} + b$ with a and b parameters to restrict the search space. These parameters are chosen large enough to include inefficient intermediate solutions that could still lead to the optimal solution, but not too large to find a solution in a reasonable amount of time (Step 4B). Accepted nodes are added to a search list. The process is repeated until all accepted nodes are searched through for the head of the decision sequence, i.e., for time layer $k \leq k_0 + K^{CTR}$. The final solution is the last updated optimal solution (result of step 3).

In Fig. 3c and d the convergence of the branch-and-bound heuristic with greedy tail is compared to the full branch-and-bound search, where not only the head but also the tail is fully optimized. The average convergence is shown for the reference of the corridor testcase (Section 2.3), with problem complexity $|M| = 4$ intersections, $|J| = 4$ movement groups per intersection, and a prediction horizon of $K = 10$ intervals with length $T = 6$ s. Both methods start in the greedy solution and use the same search space (a, b values). The heuristic method optimizes only the first two intervals, $K^{CTR} = 2$, assumes a greedy tail and reconsiders part of the greedy tail in the next solution update, whereas the full search method optimizes all intervals at once ($K^{CTR} = K = 10$). The full search method converges steadily, reaching its optimal solution level after 240 s (see Fig. 3d), however, takes too much time (slower than real-time) if an update interval of $K^{UP} = 1$ (6 s) is considered. The heuristic with greedy tail converges much faster (see Fig. 3c), within 2 s (2/12 of the available update time if $K^{UP} = 2$, and 2/6 if $K^{UP} = 1$), reaching a suboptimal solution that lies close (<1%) to the optimal solution of the full search method (see Fig. 3d). Therefore, the heuristic solution approach is efficient and accurate enough for the problem analyzed in this paper.

2.2.4. Tuning and detuning

To create the "almost perfect world" in the first step of the sensitivity analysis, the macroscopic prediction model is tuned to be equal to the deterministic microscopic driving behavior (step I in Fig. 2). This is done by measuring the values of the macroscopic parameters of model (1)-(8) in the microscopic simulation. The free-flow speed v_i [m/s] in (3) is tuned by measuring the travel time of the vehicles from link exit to link exit under free-flow conditions (and green signals). Note that an intersection has no space in the macroscopic model, and travel times on the intersection surface need to be incorporated in the (therefore slightly slower) macroscopic link speed. The saturation flow rate r_i [veh/s] in (4) is tuned by measuring the individual passage times at the link exit of vehicles leaving a queue from the moment the signal becomes green. The individual passages are averaged into the saturation flow rate r_i [veh/s]. Note that the tuned value of the saturation flow rate is slightly lower for the first intervals after turning green compared to the subsequent intervals, caused by the reaction times when switching signal states. The maximum storage capacity L_i/L_{VEH} at a link in (6) is tuned by measuring the distance headway L_{VEH} [m] in a standing queue. Note that the actual available space follows from the queueing model (7)-(8). The queue head propagation speed v^H_i [m/s] in (8) is tuned by measuring the times the vehicles start to move in the queue from the moment the signal turns green. The individual time headways are averaged and translated into a constant propagation speed (using the tuned distance headway L_{VEH} [m]). In addition to the tuning of the macroscopic model parameters, the demand input to the macroscopic prediction model is exactly known and is directly retrieved from the microscopic simulator. The individual demand pattern is recorded in advance for the entire simulation. The arrivals $a_i(k)$ [veh] in (1) at the external movements at the entrance links of the network are set equal to the (aggregated) recorded arrivals. Moreover, the turn fractions $\pi_{i\,\hat{\imath}} \in [0,1]$ in (2) and (6) are replaced by the recorded real destinations of the vehicles and become time dependent. The initial state input to the prediction model is exactly known as well and is derived from the simulation data. During the simulation, the individual vehicle passages at the link exits are measured and aggregated into the macroscopic departures $d_i(k)$, known up to time index k_0. Based on these simulation data, the initial macroscopic state of the queues $x_i(k_0)$ [veh] is calculated accordingly (1)-(3). With the perfect state initialization, exact demand pattern, and tuned model parameters, the macroscopic model can reproduce the microscopic world close enough to function as a reference in the sensitivity analysis (as will be shown in Section 3.1).

In the second step of the sensitivity analysis, the effect on the system performance of aggregation errors in the prediction model is analyzed (step II in Fig. 2). To this end, the deterministic driving behavior in the microscopic simulator is replaced by stochastic behavior (with parameters around the original average values). So, the tuned macroscopic prediction model now corresponds to the aggregated average driving behavior. Moreover, the individual demand pattern is replaced by the average demand pattern as input for the macroscopic prediction model. The arrivals $a_i(k)$ [veh] in (1) at the external movements at the entrance links of the network are set back to the average demand values. The turn fractions $\pi_{i\,\hat{\imath}} \in [0,1]$ in (2) and (6) are set back to the average fractional values instead of considering real destinations of the vehicles. The effects on the system performance of all these aggregation errors are measured.

In the third step of the sensitivity analysis, the effect of biases in the different components of the macroscopic model predictive control system on the system performance are analyzed (step III in Fig. 2). To this end, the tuned macroscopic model parameters, i.e., free-flow speed v_i [m/s] in (3), saturation flow rate r_i [veh/s] in (4), queue head propagation speed v^H_i [m/s] in (8), are biased by introducing structural errors, i.e., $\widetilde{r}_i = (1+e_r)^{*}r_i\ \forall i$, $\widetilde{v}_i = (1+e_v)^{*}v_i\ \forall i$, $\widetilde{v}^H_i = (1+e_{v^H})^{*}v^H_i\ \forall i$, with $e_r, e_v, e_{v^H} \in [-1,1]$. Additionally, the average demand pattern is biased by introducing a structural error in the average arrivals $a_i(k)$ [veh] in (1) at the network entrances, i.e., $\widetilde{a}_i(k) = (1+e_a)^{*}a_i(k)\ \forall k \forall i$, with $e_a \in [-1,1]$. Biased turn fractions $\pi_{i\,\hat{\imath}} \in [0,1]$ in (2) and (6) are considered as well, including a structural error in the main direction from movement i to downstream movement $\hat{\imath}$, i.e., $\widetilde{\pi}_{i\,\hat{\imath}} = (1+e_\pi)^{*}\pi_{i\,\hat{\imath}}$, with $e_\pi \in [-1, (1-\pi_{i\,\hat{\imath}})/\pi_{i\,\hat{\imath}}]$, and correcting the other directions to downstream movements $\bar{\imath}$, i.e., $\widetilde{\pi}_{i\,\bar{\imath}} = (1-(\pi_{i\,\hat{\imath}}/(1-\pi_{i\,\hat{\imath}}))^{*}e_\pi)^{*}\pi_{i\,\bar{\imath}}\ \forall\, \bar{\imath} \neq \hat{\imath}$, such that $\sum_{\bar{\imath}} \widetilde{\pi}_{i\,\bar{\imath}} = 1$. Finally, the initial queue states $x_i(k_0)$ [veh] in (1) are biased by introducing structural errors, i.e., $\widetilde{x}_i(k_0) = (1+e_x)^{*}x_i(k_0)\ \forall i$, with $e_x \in [-1,1]$. The effects on the system performance of these biases are measured.

Note that the performance of the control system in the microscopic world can always be measured exactly. To preserve comparability between both worlds, the microscopic system performance is expressed in macroscopic quantities, according to the macroscopic definition of queues and delay (Equations (1)-(3), with measured departures $d_i(k)$, recorded arrivals $a_i(k)$, and real destinations

for the turn fractions $\pi_{i\,i} \in [0, 1]$).

2.3. Traffic Scenario: Corridor with spillback

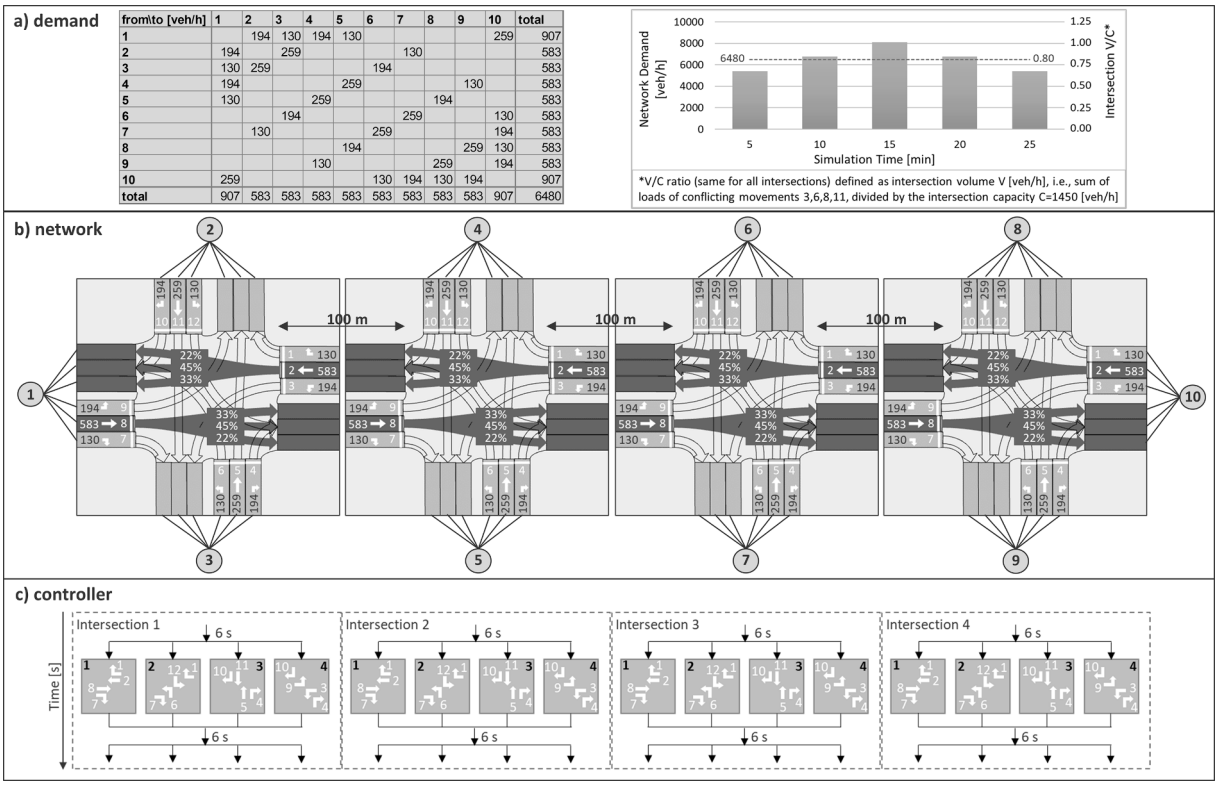
2.3.1. Network configuration

The sensitivity analysis is performed on a corridor of four intersections, where queues can cause spillback and block traffic at upstream intersections (see Fig. 4b). The intersections are connected by short low-speed links (<50 km/h), representing urban settings. The distance between the intersections is small (smaller than in real-life) to cause spillback in less simulation time and needing fewer vehicles, where a similar process will arise in real-life covering more time and involving more vehicles. Each intersection consists of the full twelve movements and considers only cars. Each movement has a separate lane, where only vehicles for this direction are allowed. Lane changes only take place at the intersections and are not allowed in the links. Vehicles only cross the intersection if there is enough space downstream and if they do not block the intersection surface. These assumptions are a simplification of reality, in which different vehicle types are mixed and driving behavior is less neat causing more incidental and severe peaks of spillback. These simplifications were done to reduce the stochastics in the sensitivity analysis experiments to create a solid reference situation ("the almost perfect world") and to isolate and focus on the effects of prediction errors.

2.3.2. Controller

The corridor is controlled by a structure-free model predictive signal controller as defined by (9). At each intersection each movement has its own signal and can be controlled separately. Signal states may switch every 6 s ($T = 6$), considering a loss time of 3 s ($T_L = 3$). At each intersection there are 4 movement groups defined of non-conflicting movements that have green at the same time (see Fig. 4c). The controller has a free choice among those 4 movement groups at each intersection, and there is no predefined order and no imposed cycle (and no other constraint on maximum waiting or green times). Moreover, the decision for a specific movement group at one intersection is independent from the choice at the other intersections, resulting in $4^4 = 256$ possible decisions in each time step of the prediction horizon (and $256^{\,K}$ possible decision sequences for the entire prediction horizon of K intervals). Note that in theory there are 8 possible movement groups per intersection (and even more when considering public transport, bikes, and pedestrians), resulting in an increasing amount of possible decision sequences. To reduce problem complexity, only the 4 most common movement groups are considered, without losing the principle of a structure-free controller.

To analyze the benefit of the prediction, the prediction horizon of the controller is increased from 6 to 120 s in the experiments ($K =$ 1,2, …,20). Note that 2 min seems enough to capture most of the traffic dynamics for this size of corridor. The controller in principle updates the decision sequence after each 12 s ($K^{UP} = 2$). However, a higher update frequency of each 6 s ($K^{UP} = 1$) is considered as well



from\to [veh/h]	1	2	3	4	5	6	7	8	9	10	total
1		194	130	194	130					259	907
2	194		259				130				583
3	130	259				194					583
4	194				259				130		583
5	130			259				194			583
6			194				259			130	583
7		130				259				194	583
8					194				259	130	583
9				130				259		194	583
10	259					130	194	130	194		907
total	907	583	583	583	583	583	583	583	583	907	6480

Fig. 4. Traffic scenario with demand (a), network configuration (b), and controller (c).

to analyze the possible reduction of prediction errors, and lower update frequencies ($K^{UP} = 5$ or 10) are analyzed for possible opposite effects. The role of the prediction horizon and the update frequency of the controller is analyzed in the sensitivity analysis under perfect as well as erroneous predictions.

2.3.3. Demand pattern

The demand pattern of the traffic entering the corridor combines different crossing traffic flows to construct spillback effects (see Fig. 4a and b). The links on the main direction consist of traffic heading in the main direction as well as traffic heading for the side directions, which can be blocked by spillback on the main direction. The intersections are equally loaded. The underlying average traffic flows are given in the OD matrix [veh/h] (Fig. 4a), and the resulting average loads [veh/h] and turn fractions are shown in the network (Fig. 4b). A total demand period of 25 min is simulated (with an additional 20 min to assure the network empties). The total demand period is split into 5 periods of 5 min each, starting in the 1st period with a demand below average, increasing in the 2nd period to the average demand level, peaking in the 3rd period at a demand above average, decreasing again in the 4th period to the average demand level, and ending in the 5th period at a demand below average (Fig. 4a). In this way, the overall demand follows a clock shape, starting at undersaturated conditions where queues easily dissolve, increasing to (over)saturated conditions where queues are building up and may cause spillback, decreasing to undersaturated conditions again where the traffic situation can recover.

In the simulation vehicle realizations are randomly generated. Each simulation is repeated 10 times with different random seeds to generalize results. These 10 random seeds remain the same for all the experiments to compare results for equal vehicle realizations. The random vehicle generation considers generation times as well as vehicle properties. The generation times between vehicles are exponentially distributed, with an average equivalent to the flow indicated in the OD matrix. The vehicle properties follow truncated normal distributions specified by the mean, standard deviation, and min–max values as given in Table 1a. Note that these vehicle properties are temporarily fixed (by setting the standard deviation to zero) in the reference situation to tune the parameters of the prediction model and to create the almost perfect world without prediction errors. The tuned parameter values of the macroscopic prediction model that correspond to the microscopic parameters of Table 1a are listed in Table 1b.

3. Results

3.1. Reference: perfect world

Before looking into the behavior of the system performance in the (almost) perfect world (step I in Fig. 2), the model match between the prediction model and the simulation model is addressed. Fig. 5a shows the simulated and predicted delay over the simulation horizon for the 10 different demand realizations. The simulation horizon of 30 min is split into intervals of 1 min, for each of which a prediction is made ($K = K^{UP} = 10$). Note that a 60-seconds prediction horizon is representative for the values used in this study and therefore long enough to check the model match. The total predicted delay for the prediction horizon (in fact the objective of the control system (9)) is compared to the total simulated delay over this 1-minute period. In this way, for each demand realization 30 predictions are made, and the model match is checked over the entire simulation period, looking into undersaturated as well as (over) saturated conditions. Fig. 5a clearly shows that the simulated delays follow the predicted delays. For each demand realization, the line of the simulated delay overlaps with the line of the predicted delay. To quantify the model match, in Fig. 5b the simulated delays are plotted against the predicted delays and the R^2-value is calculated. The points almost perfectly lie on the diagonal with an R^2-value of 0.9996, indicating an almost perfect model match.

In this almost perfect world, where accurate predictions are made, the control system can reach its optimal performance. In Fig. 6a, the system performance is presented as a function of the prediction horizon varying from 6 to 120 s (with a fixed update horizon of 12 s), and in Fig. 6b, as a function of the update horizon, varying from 6 to 60 s (with a fixed prediction horizon of 60 s). The system performance, or in fact the system cost, is expressed as the average delay per vehicle on the corridor, a more intuitive representation of the total delay objective of the controller. The average delay is obtained by dividing the total delay of the entire simulation by the total number of vehicles in the simulation, i.e., a division by a constant number, since the simulation starts and ends in an empty network, affecting only the representation scale. Note that the average delay per vehicle is presented using a log-scale to focus on the lower delay

Table 1a
Microscopic vehicle properties.

Microscopic quantity*	Mean	Standard deviation	[Min, Max]
Simulation time step [s]	0.1	–	–
Reaction time [s]	0.8	0.1	[0.6, 1.0]
Reaction time at stop/traffic light [s]	1.6	0.2	[1.2, 2.0]
Maximum acceleration [m/s^2]	3.0	0.5	[2.0, 4.0]
Normal deceleration [m/s^2]	4.0	0.5	[3.0, 5.0]
Maximum deceleration [m/s^2]	6.0	0.5	[5.0, 7.0]
Maximum speed at link [m/s]	11.1	–	–
Speed acceptance factor	1.1	0.1	[0.9, 1.3]
Clearance in queue [m]	2.0	0.5	[1.0, 3.0]
Car length [m]	4.0	0.5	[3.0, 5.0]

* The microscopic parameters follow truncated normal distributions specified by the mean, standard deviation, and min–max values.

Table 1b
Tuned macroscopic quantities.

Macroscopic quantity		Tuned value
T	Time step [s]	6.0
T_L	Loss time [s]	3.0
r_i	Saturation flow rate [veh/s] *	0.5
v_i	Speed at dedicated lane [m/s] **	8.3
L_i	Length of dedicated lane [m]	100.0
L_{VEH}	Distance headway in queue [m]	6.0
L_i/L_{VEH}	Storage capacity of lane [veh]	16.0
v_i^H	Propagation speed queue head [m/s]	4.0

* Note that the saturation rate is adjusted to 0,33 veh/s for the first two time steps after green, including the effect of reaction times.

** Note that the tuned macroscopic free-flow speed is lower than the maximum speed in the micro simulation, incorporating the travel time on the intersection surface. Note that a vehicle needs 12 s, i.e., $n_i = (L_i/v_i)/T = 2$ intervals, to travel from one intersection to the next in free-flow conditions.

values. The results of the 10 different simulation realizations are averaged, and the 95% confidence intervals are calculated to indicate the variation in the simulation results (together with min–max bandwidths that are presented on the background). Pairwise *t*-tests are performed to indicate whether the system costs are significantly changing for an increasing prediction horizon or update horizon, where a probability below 5% indicates a significant increase or decrease.

Fig. 6a shows that increasing the prediction horizon improves the system performance in the corridor when perfect predictions are available. A short prediction horizon (6 s) considers only local traffic information nearby each individual intersection. This causes spillback and results in a high average delay of 313 [s] per vehicle. Increasing the prediction horizon allows looking ahead over multiple intersections, reduces spillback effects and decreases system costs up to an average delay of 85 [s] per vehicle. Note that the delay is not a smooth and strictly decreasing function. Fig. 6a shows that there is a considerable bandwidth around the average results. The delay curve for each of the simulated realizations is fluctuating over the prediction horizons, particularly at the beginning of the curve for short horizons. Due to the finite nature of the prediction horizon and the inability of too short horizons to capture the whole process dynamics, each prediction horizon length results in a different sequence of optimization problems, a different sequence of optimal control plans and traffic states, and therefore in a different system performance. Moreover, the delay also varies considerably over the different demand realizations, causing spillback at different moments and with different severity. This is inherent to the (over) saturated traffic scenario with a high probability for spillback and due to the stochastic generation of individual vehicles. Averaging the realizations leads to less fluctuation in the system costs and a smoother curve. Pairwise *t*-tests indicate that the delay is a (piecewise) decreasing function (see Fig. 6a). The delay significantly decreases with a large amount up to a horizon of 24 s, i.e., looking ahead 2 intersections considering a travel time of 12 s between intersections (Table 1b). The delay remains almost equal for a horizon of 30 s, and significantly decreases again, however with a smaller amount, for a horizon of 36 s, looking ahead 3 intersections. The system performance does not significantly improve anymore beyond this horizon for this corridor of 4 intersections.

Fig. 6b shows that updating the decision more frequently using a shorter update horizon improves system performance only slightly when almost perfect predictions are available. A shorter update horizon results in more overlap in the subsequent decision sequences, shifting the prediction horizon more frequently with smaller steps, and gives more steady predictions and a better performing control system. When there is no overlap at all and the update horizon equals the prediction horizon of 60 s, consequences of decisions at the end of the update horizon are no longer considered, and the system costs significantly increase, however only gradually in absolute sense (increase in average delay from 85 [s] to 97 [s] per vehicle). The predictions are (almost) perfect and do reproduce the traffic situation well enough for longer prediction horizons, keeping the system performance steady, also for longer update horizons.

3.2. Aggregation errors

Aggregating individual behavior in the prediction model (step II in Fig. 2) introduces a model mismatch. Fig. 5c shows that the line of the simulated delay no longer overlaps with the predicted delay, but that simulated delays deviate from predicted delays. However, the pair of lines that belong to the same run of prediction and simulation still lie close to each other and can be visually matched. The R^2 analysis in Fig. 5d shows that the points no longer lie on the diagonal but are more scattered and lie slightly skew to the diagonal (the R^2 value decreases to 0.9846). The prediction model slightly underestimates the delay by assuming aggregated and uniformly distributed behavior and makes control decisions accordingly. In the simulation (as in real-life) platoons of vehicles are formed, consisting of clustered vehicles heading for the same direction or arriving at the same time. These platoons do not match the control decisions based on aggregated uniform behavior and therefore lead to a higher and more varied delay than predicted.

Despite the aggregation error in the prediction model, the predictive properties of the control system remain preserved. Fig. 6c shows that the system costs still decrease, i.e., system performance still increases, for increasing prediction horizon. However, there is a significant performance loss compared to the reference situation with perfect predictions (see *t*-values in Fig. 6c). The system cost increase from an average delay of 85 [s] to 105 [s] per vehicle. From a more detailed analysis it appears that 85% of the performance loss is caused by aggregating the individual destinations of vehicles into turn fractions, 10% is caused by aggregating individual arrivals into average demand, and 5% is caused by aggregating individual driving behavior. The aggregation into turn fractions is

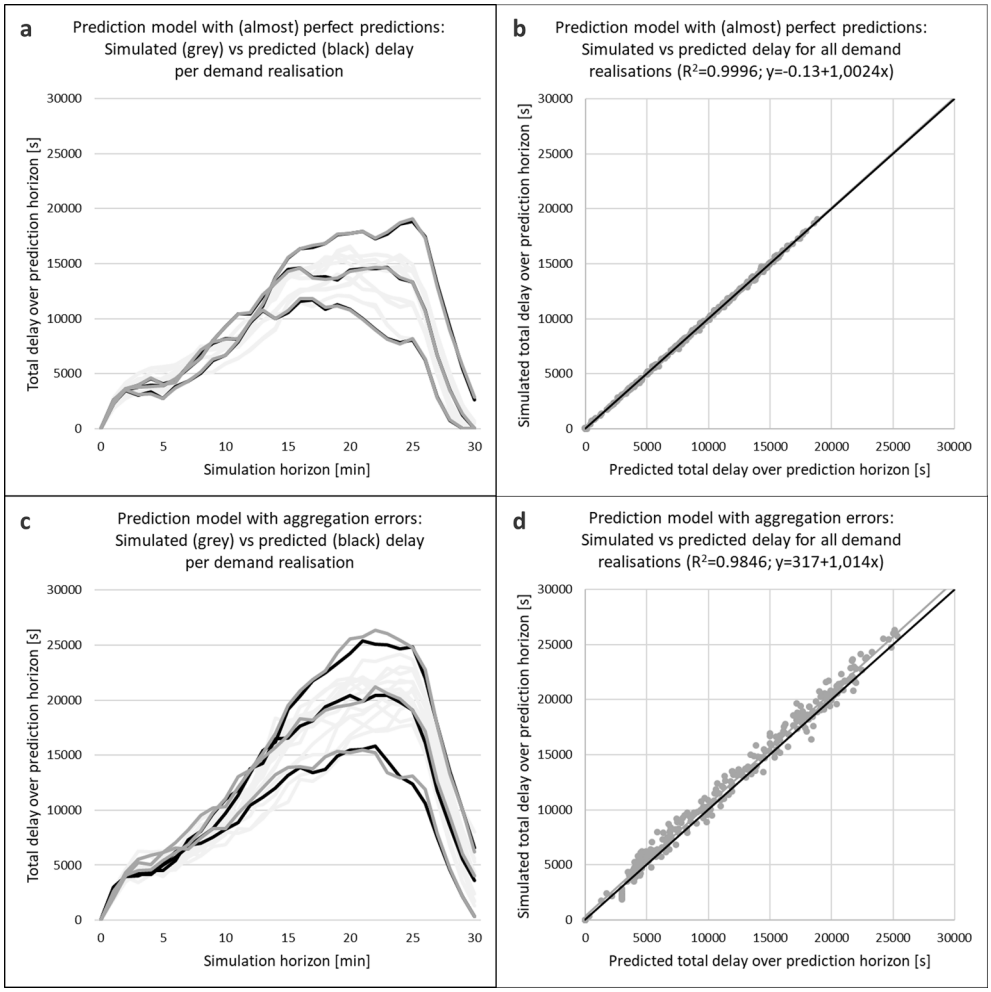


Fig. 5. Model match between the microscopic simulation model and macroscopic prediction model for the reference 'almost perfect world' (top) and the presence of aggregation errors (bottom). The model match for all 10 simulation realizations is presented, in total as a quantitative R^2 analysis (right), and per realization as a function over the simulation horizon (left) where 3 realizations are highlighted.

dominant for the simulated corridor, where many platoons are formed consisting of vehicles heading for the same direction. Platoons caused by clustered arriving vehicles (demand), or slower vehicles (driving behavior) play a lesser role in the simulated corridor, since the intersections lie close to each other. Moreover, these platoons affect the arriving times of vehicles in undersaturated conditions, and therefore have less influence in the simulated corridor with its many queues. Aggregated turn fractions cause the largest performance loss in saturated conditions. Accurate predictions of the individual turning directions of vehicles are most important in situations where the system is close to spillback from downstream movements. In these critical moments the aggregated prediction model with turn fractions is not accurate enough and causes significant performance loss.

The performance loss due to aggregation errors is clearly visible at large update horizons (see Fig. 6d). For an update (and prediction) horizon of 60 s the system costs increase from an average delay of 97 [s] to 142 [s]. Note that for these settings the model mismatch is displayed in Fig. 5. The performance loss can partly be compensated by decreasing the update horizon and reinitializing to the current state more often. However, a significant performance loss remains (see *t*-values in Fig. 6d), leading to an increase in the average delay of 85 [s] to 105 [s] per vehicle. So, aggregating individual vehicle behavior in the prediction model results in a significant performance loss, or reasoning the other way around, including individual vehicle information in the predictive controller results in a considerable performance gain (provided that accurate predictions are available for individual vehicles, their destinations in particular).

3.3. Biases

Table 2 presents the system performance of the controller when there are additional biases in the aggregated prediction model (step III in Fig. 2). For the different model quantities, the effect on the system costs of a range of relative biases is shown, i.e., $e = -0.5, -0.2,$

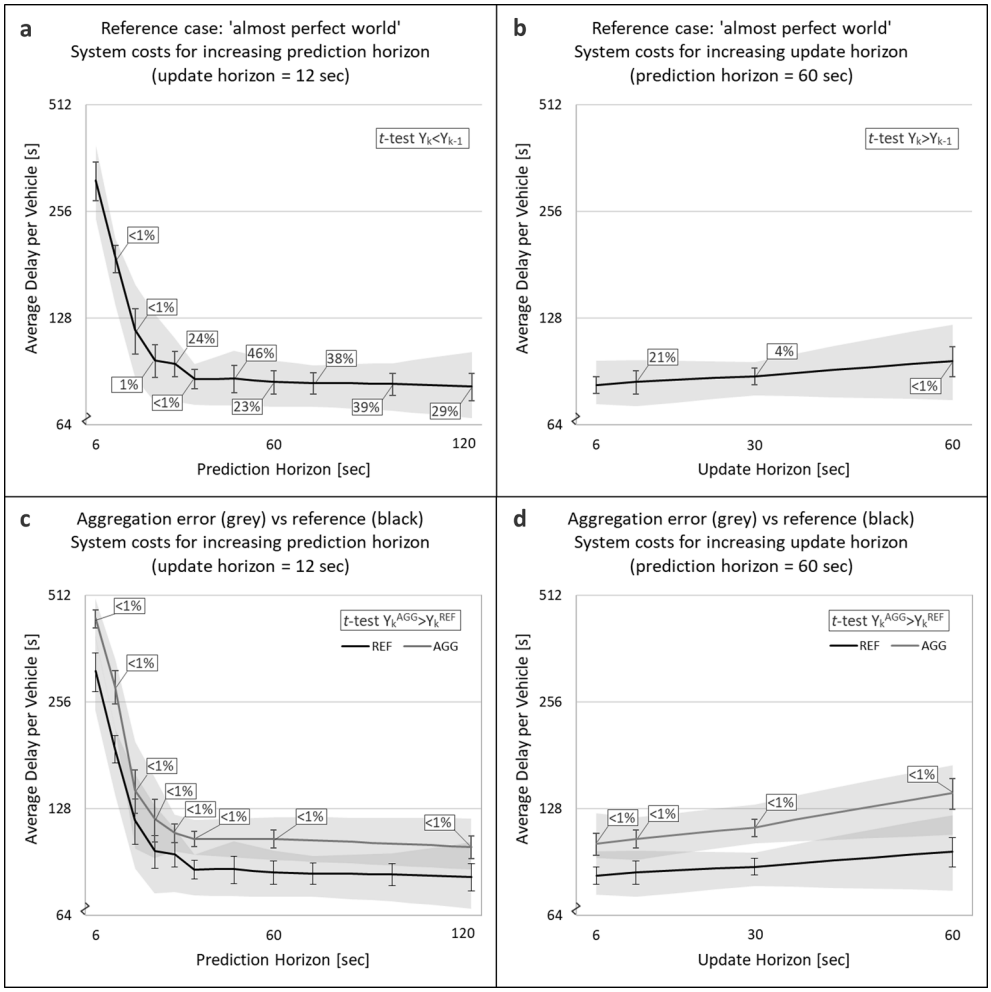


Fig. 6. Performance of the controller with almost perfect predictions (top) and aggregation errors (bottom) for an increasing prediction horizon (left) and update horizon (right). Average values over 10 simulation realizations are presented, together with 95% confidence intervals (and min–max bandwidths on the background) to indicate variation, and *t*-values to indicate significant changes.

−0.1, 0.1, 0.2, 0.5. Note that for each bias (table row) the average delay per vehicle can be graphically displayed as a function of the prediction horizon and as a function of the update horizon, in a similar way as for the reference and aggregation error in Fig. 6. To get a total overview of the effect of all the biases, the essential points of the delay curves are listed in the table instead and the cell of each delay is colored according to its value. For comparison purposes also the system costs for the reference and aggregation error are listed (as displayed in Fig. 6).

3.3.1. Role of the prediction horizon

Although there is an added bias in the aggregated prediction model, the predictive properties of the control system remain preserved. The system costs still decrease for increasing prediction horizon (colors change from dark to light for increasing prediction horizon in Table 2). However, for some quantities, like turn fractions and saturation rate, for the larger biases of 50%, the prediction horizon needs to be increased to 60 s instead of 30 s to fully benefit from the lookahead capability. Note that in the experiments only one quantity is biased at a time and the other quantities contain accurate information. Apparently, despite the biases, there is enough structure in the predictions left to be beneficial for the control system. Overall, looking ahead and exchanging information from multiple intersections, even if the information contains biases that propagate through the network, leads to a better performance than considering only local information. Connecting intersections is of major importance at near saturated conditions to reduce the chance of spillback and improve the dissolving of queues.

3.3.2. Role of the update horizon

The update frequency plays an important role in reducing the performance loss caused by biases. Shortening the update horizon decreases the system costs (colors change from dark to light for shorter update horizons in Table 2). More frequently reinitializing to

the actual state reduces the performance loss caused by biases. The reducing effect differs per quantity and error level. However, the performance loss is not compensated completely, i.e., if there is a performance loss due to the bias at an update horizon of 60 s, then there remains a performance loss at an update horizon of 12 s. Moreover, if the state itself is biased too much, the reinitialization to the state has no compensating effect anymore and/or the biased state itself causes an additional performance loss (see the −50% state bias in Table 2 where the performance loss due to aggregation errors is no longer reduced by shorter update horizons and the 50% state bias where an additional performance loss is measured due to the state bias on top of the performance loss due to aggregation errors).

3.3.3. Effect of a bias on system performance

In general, the biases in the prediction model result in a loss of system performance. The average delay per vehicle for the control system with biased predictions is mostly higher than for the control system with aggregation errors only (see Table 2, column for prediction horizon 60 s and update horizon 12 s). Sometimes there is no significant performance loss for the smaller error levels of ± 10% and ± 20%. Sometimes part of the performance loss caused by the aggregation error can even be compensated by introducing an

Table 2
System costs expressed in average delay per vehicle [s] for the predictive controller with biases, as a function of the prediction horizon (with fixed update horizon of 12 s) (left) and as a function of the update horizon (with fixed prediction horizon of 60 s) (right).

Legend													
Average delay per vehicle [s]	≤90	≤100	≤110	≤120	≤130	≤140	≤150	≤200	≤250	≤300	≤400	≤500	≤600

			prediction horizon [s]			update horizon [s]		
			12	30	60	12	30	60
Reference			189	95	85	85	88	97
Aggregation			284	110	105	105	113	142
Bias	Turn fractions (main direction) $\pi_{ii} \in [0,1]$	-100%	271	109	102	102	123	163
		-50%	236	105	100	100	108	142
		-20%	267	107	104	104	107	132
		20%	325	127	114	114	128	161
		50%	368	158	129	129	145	200
		100%	396	286	234	234	301	353
	Queue head propagation speed v_i^H [m/s]	-75%	303	113	117	117	151	291
		-50%	314	118	118	118	140	159
		-20%	307	115	108	108	120	137
		20%	310	112	103	103	112	133
		50%	324	110	105	105	108	120
		75%	319	104	108	108	109	126
	Saturation flow rate r_i [veh/s]	-50%	607	461	275	275	314	416
		-20%	396	167	142	142	148	166
		-10%	323	127	116	116	127	157
		10%	224	103	100	100	107	127
		20%	190	103	102	102	108	131
		50%	210	118	115	115	124	156
	Travel time L_i/v_i [s]	-50%	292	167	149	149	146	155
		-20%	284	110	105	105	113	142
		-10%	284	110	105	105	113	142
		10%	284	110	105	105	113	142
		20%	284	110	105	105	113	142
		50%	326	105	100	100	106	141
	Demand Arrivals $a_i(k)$ [veh]	-50%	190	97	91	91	103	122
		-20%	202	103	99	99	107	120
		-10%	234	110	104	104	109	121
		10%	336	118	112	112	122	148
		20%	370	127	118	118	129	161
		50%	491	157	146	146	155	281
	State Queue $x_i(k_0)$ [veh]	-50%	248	153	139	139	132	136
		-20%	192	114	107	107	111	123
		-10%	263	110	100	100	110	133
		10%	332	118	106	106	121	150
		20%	339	118	111	111	122	149
		50%	414	174	141	141	142	198

additional bias, resulting in lower average delays. For example, an overestimation of the travel time partly compensates the aggregation error (see + 50% travel time bias in Table 2). By aggregating the individual vehicle behavior, all vehicles are assumed to have the same speed in the prediction model. Consequently, the slowest vehicles miss the predicted green window and need to be given green twice, resulting in a performance loss caused by the aggregation error. By overestimating the travel time in the prediction, all vehicles become slower in the prediction model, and fewer vehicles will miss the predicted green window, re-gaining some of the performance. (Note that in this way there will be more vehicles that arrive slightly before the green window, but this delay is much smaller than when the window is missed.) Similar compensating effects of the performance loss due to aggregation errors can be noticed for other quantities. However, note that the average delays remain always higher than that for the reference situation, hence there is always a performance loss by introducing a bias compared to the almost perfect world without prediction errors.

For all quantities, a bias in one direction has a larger negative effect on the system performance than a bias in the opposite direction (colors are darker in one bias direction than in the opposite direction in Table 2). For example, an underestimation of the saturation rate causes a large performance loss, since the controller predicts too long and superfluous green periods to serve the traffic, reducing the capacity of the system. An overestimation of the saturation rate has less impact on the system performance. Since too short green periods are predicted, no predicted green is really wasted but given to other movements (and there is only a small capacity reduction by switching between movements more often). A small overestimation of the saturation rate may even be beneficial for the control system in (over)saturated conditions, since shorter green periods will be given to crossing movements, and the main direction on the corridor will get more gaps to dissolve its queues and reduce its spillback, compared to the control system with aggregated demands. Note that in undersaturated conditions this benefit disappears and there is a performance loss due to too short green periods and switching too often between movements. Similar effects can be noticed for the other quantities: the bias direction that clearly reduces the capacity of the control system causes the largest performance loss, and the other bias direction has less impact and may even be beneficial (compensate aggregation errors) for the control system in saturated conditions.

3.3.4. Ranking of quantities

Comparing all quantities, the saturation rate has the largest effect on the system performance in the simulated saturated corridor. For the smaller biases (up to −20%) the saturation rate already causes a significant performance loss, that cannot be compensated that well by reinitialization to the state more frequently. For larger biases (-50%) in the saturation rate the performance loss is clearly higher than for the other quantities. The high sensitivity for the saturation rate was also found in the authors' earlier work (Poelman et al., 2020) for a single intersection. In a saturated corridor, inaccurate turn fractions also have a large effect on the system performance (note it was already the dominant factor for the performance loss due to aggregation). For a small additional bias (up to 20%), there already is a significant performance loss, however, the performance loss can be reduced quite well by reinitialization to the actual state. Accurate turn fractions are important at critical situations where the system is close to spillback from downstream movements. Reinitializing to the actual queue state more often, helps in predicting the available space downstream in these critical moments. For large biases (+100%) in the turn fractions, i.e., almost all traffic (twice the original 45% resulting in 90% of the traffic) heads for the main direction in the prediction (not an unrealistic assumption if direction information is missing), the prediction errors are too large to compensate and cause a huge performance loss. The biased demand, a structural error in the number of arriving vehicles, has less impact in the saturated corridor, since there are many queues in the network. The effect only becomes noticeable for larger biases (50%), when the predicted amount of crossing traffic becomes too large to find enough possibilities for the main direction of the corridor to dissolve its queues and reduce its spillback. Moreover, the demand bias can be compensated quite well by reinitialization to the actual queue state. The compensation between demand and queue information was also noticed in the authors' earlier work (Poelman et al., 2020) for a single intersection. The travel time (or speed) has a relatively small effect in the saturated corridor. It becomes a more relevant quantity in undersaturated conditions in a network with larger distances between the intersections. The queue propagation speed has the smallest effect. Biases in the queue propagation speed cause temporary errors in the predicted storage space, that can be corrected well by reinitializing to the actual queue states. Only larger biases lead to a significant capacity reduction and performance loss. Considering this ranking, note that the state is still the most important quantity of them all, since it is needed to reduce the effect of the biases of the other quantities.

4. Discussion

The results of the sensitivity analysis show that predicting, by looking ahead over a horizon that corresponds to the propagation of traffic over multiple intersections, improves the system performance of the analyzed structure-free model predictive controller. This happens not only when perfect predictions are available but also when there are aggregation errors and biases in the system. This demonstrates the advantage of a global predictive network controller compared to local control in saturated networks. However, the sensitivity analysis also shows that there are conditions and guidelines that should be considered to take full advantage of the systems benefits. These will be discussed below.

The sensitivity analysis results show that increasing the prediction horizon increases system performance, in perfect conditions without prediction errors as well as in erroneous conditions with aggregation errors and biased predictions. There is an optimal horizon beyond which system performance does not improve anymore. This optimal horizon may be larger for a system containing biases, to fully benefit from the lookahead capability. In predictive control applications, a prediction horizon should be chosen that is large enough to look ahead in downstream (and upstream) direction over at least two intersections of a corridor, since connecting information on intersections is essential to improve system performance in saturated networks with spillback. The prediction horizon does not necessarily have to include (the traffic propagation over) all intersections on a corridor, since the performance gain becomes

relatively smaller with the addition of each extra intersection. The exact choice of the prediction horizon can be expected to depend on the network configuration and should be larger if the distance between intersections is larger. However, a similar relation between system performance and increasing prediction horizon is expected as for the studied corridor.

The sensitivity analysis also shows that increasing the update frequency, which is the same as reinitializing to the current state more often, increases system performance and significantly reduces the effect of aggregation errors and biases in the control system. In structure-free predictive control applications, the update horizon should be set as small as possible (in units of seconds) to benefit from the highly adaptive property of the structure-free controller to correct mistakes quickly. Moreover, the accuracy of the initial state is essential to reduce the effect of prediction errors. If the initial state is biased too much, the performance gain of reinitializing to the state may be lost, and the effect of prediction errors may no longer be counteracted, as shown in the sensitivity analysis where aggregation errors are no longer compensated by reinitialization to a biased state. Therefore, the initial state is most important to predict accurately in the structure-free predictive control system.

In real life applications there is always a trade-off between the choice of the update horizon and the choice of the prediction horizon. If the update horizon is shortened to improve system performance by reinitializing to the current state more often, less calculation time is available to find an optimal control plan for the upcoming prediction horizon, for real-time operation. The prediction horizon should be decreased to limit the search space and to find an optimal solution in time, which may reduce the performance gain of the shortened update horizon. Or stated the other way around, if the prediction horizon is increased to improve system performance, more calculation time is needed to find an optimal control plan. So, the update horizon should be set larger, which may reduce the performance gain of the increased prediction horizon. This trade-off should be considered in the control application, depending on the available computational resources, particularly in large networks. In the studied corridor, however, this was not a dominant factor yet.

From the sensitivity analysis it turns out that aggregation errors in the prediction model cause a significant performance loss. The performance loss is expected to be even higher in real life, since there is more randomness in real life than in the simulation environment considered in this study. Note that the performance loss may be less for the more traditional cycle-based controllers, compared to the structure-free controller considered in this study, since cycle-based controllers make more aggregated decisions based on proportions of traffic. If a structure-free controller is used, individual information should be used, to fully benefit from the adaptive potential of the structure-free controller and significantly increase system performance. However, this individual vehicle information should be predicted carefully, otherwise biases can decrease performance again below the level of an aggregated system. The performance of future control applications can be improved significantly by using a structure-free controller and including individual vehicle information, in particular individual destinations, if the individual information is predicted accurately.

The sensitivity analysis shows that in general biases cause a significant additional performance loss (on top of the performance loss due to aggregation errors). Smaller biases (up to 20%) cause a proportional performance loss (up to 25%), which can partly be compensated by reinitializing to the actual state. Larger biases (50%) may result in a non-proportional performance loss (up to 150%), which can hardly be compensated by reinitializing to the actual state more frequently. Such large biases are not common in practice, except for quantities that are difficult to predict and are based on rough assumptions. An example that could occur is a lack of information on individual vehicle destinations or turn fractions for which it is assumed that all traffic is heading for the main direction. These rough assumptions resulting in large biases should be avoided. Note that in the sensitivity analysis the quantities have been disturbed one by one. In real life, biases of different quantities appear simultaneously and may amplify each other, resulting in more performance loss than indicated in this study. In the sensitivity analysis, a small bias could already cause significant performance loss for quantities that cannot be corrected by the initial state (for smaller update horizons) or by information on other quantities in the system (for larger prediction horizons). These model quantities are most important to predict accurately.

From the sensitivity analysis results, the saturation rate and the turn fractions appear to be the most sensitive quantities in the predictive control system and need to be predicted most accurately. However, these quantities are also most difficult to predict in real life, by lack of information and due to platooning (for example a slow vehicle decreasing the saturation rate for a whole platoon of following vehicles, or a platoon of vehicles with the same destination disturbing the average turn fractions). In the design of the control system, the prediction methods for these quantities, i.e., saturation rate and turn fractions, need to be improved and need to be made time-dependent, particularly in combination with individual vehicle information. Note that the quantities addressed in the sensitivity analysis in this paper are the major components in a prediction model. A different macroscopic traffic flow model will change the exact results of the experiments but is expected to identify comparable model components as the most sensitive quantities in the control system. Note that some of the quantities, like travel time and queue head propagation speed may have more influence in network configurations with longer distances between intersections and other demand patterns. However, the saturation rate and the turn fractions are expected to remain the most sensitive quantities in saturated conditions since these quantities are essential in critical situations where the system is close to spillback.

For all quantities investigated in the sensitivity analysis, a bias in one direction has a larger negative effect on the system performance than a bias in the opposite direction. The bias direction that reduces the capacity of the intersections has the largest effect on the system performance in (over)saturated conditions. This bias direction should be avoided for all quantities, particularly if biases in different quantities amplify each other. In the design towards a more robust controller, one should slightly over/underestimate each quantity to the direction with the least impact on system performance. Moreover, adding a small deliberate bias in the least impact direction may even be beneficial for the control system, partly regaining some of the performance loss due to aggregation errors. As shown in the sensitivity analysis, adding a small bias to some of the quantities (such as the saturation flow) leads to a better performance than just taking the measured average. In (over)saturated conditions a small bias in a quantity in the right direction helps dissolving queues and reducing spillback, however, such a preventive bias may introduce an additional performance loss in

undersaturated conditions. In the design of the control system, it should be considered how much performance loss is acceptable in many common situations compared to the huge amount of performance loss in fewer exceptional situations. This consideration deserves further research.

5. Conclusion

In this paper a sensitivity analysis was set up and performed for a structure-free model-based predictive signal controller in a saturated urban corridor with spillback. In a simulation environment, the influence of prediction errors was studied on the system performance of the controller, as a function of the prediction horizon (lookahead capability) and update frequency (damping ability) of the control system. The results of the sensitivity analysis were translated into guidelines for the design of a structure-free model-based predictive controller, such that the highly adaptive and predictive system can better handle the effects of prediction errors in an (over) saturated network.

From the sensitivity analysis it can be concluded that the predictive property of the control system is strong and remains preserved under erroneous conditions. Increasing the prediction horizon from 0 to 60 s increases system performance up to 75% in the corridor when perfect predictions are available, but also when there are aggregation errors or even biases in the system, leaving enough remaining structure in the prediction model to rely on. This demonstrates the advantage of a global predictive network controller compared to local control in saturated networks. Looking ahead connects information of multiple intersections on arriving traffic (forward) and spillback (backward). The performance gain increases but flattens out with an increasing number of connected intersections. The gain is 40%, 70%, and 75% when looking ahead 1, 2, and 3 intersections in the simulated corridor. There is an optimal prediction horizon, beyond which the system performance does not improve anymore, however this horizon is not always easy to recognize due to the highly stochastic traffic process and may be longer for a system containing biases. This optimal choice of the prediction horizon is network dependent, but the behavior of the prediction horizon is expected to be similar as in the simulated corridor.

The update frequency plays an important role to reduce the effect of aggregation errors and biases. Decreasing the update horizon from 60 to 12 s in the corridor increases system performance up to 30%. Due to the highly adaptive property of the structure-free controller, mistakes can be quickly corrected by reinitializing to the actual traffic state more often. The sensitivity analysis has shown that this correcting ability can disappear if the initial state estimate contains biases, therefore the initial state is most important to estimate accurately in the structure-free predictive control system.

Although partly reduced, there remains a significant performance loss due to aggregation errors and biases. Aggregating individual vehicle behavior leads to a significant performance loss of 20% in the corridor, mainly caused by aggregating individual destinations into turn fractions, in a lesser extent also by aggregating individual arrivals and driving behavior. The latter may be larger in real-life due to more stochastics and larger intersectional distances. A structure-free controller can adapt the control decision to closely match fluctuating arrival pattens and platooning. Therefore, there is a large benefit by including more detailed information on individual vehicle arrivals, in particular individual destinations, into the prediction model, provided that this type of information is available in real-life and can be predicted accurately enough.

Additional performance losses due to biases were detected for all model quantities in the sensitivity analyses. Smaller biases (up to 20%) in the model quantities lead to proportional performance losses (up to 25%) in the corridor. Larger biases (of 50%) may result in an unproportioned performance loss (up to 150%) in the corridor and should therefore be avoided. The saturation rate is the most sensitive quantity in the control system in the saturated corridor, followed by the turn fractions already indicated as the dominant factor in the aggregation process. A bias in the saturation rate already causes a significant performance loss at the lower error levels that is difficult to compensate by reinitialization to the actual state. Therefore, the saturation rate is the most important model quantity to predict accurately.

The sensitivity analysis also shows that for all model quantities one direction of the bias has more impact on the system performance than the other direction. The bias direction that reduces the capacity of the intersections has a large effect on the system performance in (over)saturated conditions and should be avoided for all quantities. A bias in the other direction is less severe and (in small amount) may even be beneficial to the system, regaining some (up to 10%) of the performance loss due to aggregation errors. In (over)saturated conditions a small bias in a quantity in the right direction helps dissolving queues and reducing spillback, however, may introduce an additional performance loss in undersaturated conditions. This consideration is left for further research towards more robust control systems.

Overall, considering these insights and guidelines, a structure-free model-based predictive controller can be designed to function adequately under erroneous conditions. Moreover, the adaptive ability makes the structure-free predictive controller promising for future applications, where information on individual vehicles becomes available.

CRediT authorship contribution statement

M.C. Poelman: Conceptualization, Methodology, Software, Investigation, Validation, Visualization, Writing – original draft. **A. Hegyi:** Conceptualization, Methodology, Validation, Supervision, Writing – review & editing. **A. Verbraeck:** Methodology, Validation, Supervision, Writing – review & editing. **J.W.C. van Lint:** Methodology, Validation, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is sponsored by the Dutch Foundation for Scientific Research NWO / Applied Sciences under grant number 16270 under project title “MIRRORS - Multiscale integrated traffic observatory for large road networks”.

References

Aboudolas, K., Papageorgiou, M., Kosmatopoulos, E., 2009. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. Transp. Res. Part C: Emerg. Technol. 17 (2), 163–174.

Astarita, V., Giofre, V.P., Guido, G., Vitale, A., 2017. The use of adaptive traffic signal systems based on floating car data. Wirel. Commun. Mob. Comput.

Burger, M., van den Berg, M., Hegyi, A., De Schutter, B., Hellendoorn, J., 2013. Considerations for model-based traffic control. Transp. Res. Part C: Emerg. Technol. 35 (October), 1–19.

Li, L., Wen, D., Yao, D., 2014. A survey of traffic control with vehicular communications. IEEE Trans. Intell. Transp. Syst. 15 (1), 425–432.

Lin, S., De Schutter, B., Xi, Y., Hellendoorn, H., 2012. Efficient network-wide model-based predictive control for urban traffic networks. Transp. Res. Part C: Emerg. Technol. 24 (October), 122–140.

Aimsun, 2017. Aimsun 8.2 User’s Manual, Aimsun Version 8.2.0, Barcelona, Spain. Available from (in software): <qthelp://aimsun.com.aimsun.8.2/doc/UsersManual/TheAimsunEnvironment.html>.

Klunder, G., Taale, H., Kester, L., Hoogendoorn, S., 2014. The effect of inaccurate traffic data for ramp metering: comparing loop detectors and cameras using information utility. In: Proceedings of the 19th IFAC World Congress, vol. 47(3), pp. 11318–11325.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y., 2003. Review of road traffic control strategies. Proc. IEEE 91(12), 2043–2067.

Poelman, M.C., Hegyi, A., Verbraeck, A., van Lint, H., 2020. Sensitivity analysis to define guidelines for predictive control design. Transp. Res. Rec. 2674 (6), 385–398.

Tettamanti, T., Varga, I., Péni, T., Luspay, T., Kulcsár, B., 2011. Uncertainty modeling and robust control in urban traffic. In: Proc. 18th IFAC World Congress, vol. 44 (1), pp. 14910−14915.

Tettamanti, T., Luspay, T., Kulcsár, B., Péni, T., Varga, I., 2014. Robust control for urban road traffic networks. IEEE Trans. Intell. Transp. Syst. 15 (1), 385–398.

van Katwijk, R.T., 2008. Multi-Agent Look-Ahead Traffic-Adaptive Control. TRAIL Thesis Series T2008/3, The Netherlands TRAIL Research School, Delft.

van Lint, H., Hinsbergen, C., 2012. Short-term traffic and travel time prediction models. In: Transportation Research Circular E-C168, Artificial Intelligence Applications to Critical Transportation Issues.

Vlahogianni, E., Karlaftis, G., Golias, J., 2014. Short-term traffic forecasting: Where we are and where we’re going. Transp. Res. Part C: Emerg. Technol. 43 (1), 3–19.

Wu, X., Liu, H.X., 2014. Using high-resolution event-based data for traffic modeling and control: an overview. Transp. Res. Part C: Emerg. Technol. 42 (May), 28–43.

Ye, B.-L., Wu, W., Gao, H., Lu, Y., Cao, Q., Zhu, L., 2017. Stochastic model predictive control for urban traffic networks. Appl. Sci. 7 (6), 588.

Ye, B.-L., Wu, W., Ruan, K., Li, L., Chen, T., Gao, H., Chen, Y., 2019. Survey of model predictive control methods for traffic signal control. IEEE/CAA J. Autom. Sin. 6 (3), 623–640.