Research report

Adaptive anticipatory network traffic control using iterative optimization with model bias correction

Wei Huang a, Francesco Viti b, Chris M.J. Tampère a

a Leuven Mobility Research Center, CIB, KU Leuven, Belgium

b Faculty of Science, Technology and Communication, University of Luxembourg, Luxembourg

This research work has been submitted to Transportation Research Part C. If you want to cite this report, please use the following reference instead:

Huang, W., Viti, F., Tampère, C.M.J., 2015. Adaptive anticipatory network traffic control using iterative optimization with model bias correction. (submitted to Transportation Research Part C)

KU Leuven
Leuven Mobility Research Center, CIB
Celestijnenlaan 300A – PO Box 2422
3001 Heverlee (Belgium)
Email: wei.huang@kuleuven.be

www.kuleuven.be/traffic
Adaptive anticipatory network traffic control using iterative optimization with model bias correction

Wei Huang a*, Francesco Viti b, Chris M.J. Tampère a

a Centre for Industrial Management/Traffic and Infrastructure, KU Leuven, Belgium
b Faculty of Science, Technology and Communication, University of Luxembourg, Luxembourg

* Corresponding author.
E-mail: wei.huang@kuleuven.be (Wei Huang)
Centre for Industrial Management, Traffic and Infrastructure, KU Leuven, Celestijnenlaan 300A, PO Box 2422, 3001 Heverlee, Belgium
Tel.: +32 16 37 27 69

Abstract: Anticipatory signal control in traffic networks adapts the signal timings with the aim of controlling the resulting (equilibrium) flow patterns in the network. This study investigates a method to support control decisions for successful applications in real traffic systems that operate repeatedly from day to day. A main bottleneck in designing the daily control scheme is the presence of model uncertainty. Conventional methods adopt a two-step procedure, iteratively updating parameter estimation and control optimization. Inconsistency arises due to the inevitable structural model-reality mismatch. This paper proposes an iterative optimizing control method to tackle this limitation and drive the traffic network towards the true optimal performance despite of model uncertainty. This Iterative Optimizing Control with Model Bias Correction (IOCMBC) corrects model bias using measurements and the resulting reality-tracking metamodel is updated for the subsequent control optimization. Theoretical analysis on matching between the IOCMBC optimal solution and the true optimum is presented. A local convergence analysis is also elaborated to investigate conditions required for a convergent scheme. One critical issue is the involvement of the sensitivity (Jacobian) information of the real route choice behavior with respect to signal control variables. To avoid performing additional perturbations, we introduce a measurement-based implementation method for estimating the operational Jacobian that is associated with the reality. Numerical tests in a small network verify the effectiveness of the proposed IOCMBC method in tackling model uncertainty, as well as a practical setting for regulating the reality-tracking convergence.

Keywords: anticipatory traffic control, adaptive signal control, model uncertainty, iterative optimization, model bias correction

1 Introduction

Traffic control and management strategies offer high opportunity to improve traffic operations. Incorporating travelers’ response is crucial to design appropriate strategies. In particular, a flow anticipatory control is proposed (Taaale, 2008), whereby the controller anticipates route choice response, thus has the possibility to look for signal settings that correspond to optimal response from the travelers in terms of e.g. optimal travel time. In game-theoretical terms, anticipatory control
recognizes control optimization as a leader’s role in the Stackelberg-type interaction between route choice (traffic assignment) and control. As it suggests a global optimization scheme with respect to the user-optimal assignment, anticipatory control is also known as bi-level optimization or global optimization problem, in terms of signal control optimization in literature (Yang and Yagar, 1995; Yang and Bell, 1998; Cascetta et al., 2006; Cantarella et al., 2012).

For transportation planning purposes, flow anticipatory control is an effective way to guide a traffic network to more desirable equilibrium conditions by setting appropriately the traffic signals. From a perspective of operational level, the calculation of anticipatory signal settings can be interpreted as determining a responsive signal control policy that is employed in daily repeated traffic operations (Smith and Van Vuren, 1993; Hu and Mahmassani, 1997). This anticipatory control scheme usually locates at a strategic level in many adaptive control systems with a hierarchical structure such as UTOPIA (Mauro and Di Taranto, 1990) and RHODES (Head et al., 1992). Signal timings operating at the strategic level are designed to react to traveler route choice response in a longer temporal and spatial horizon, and could be subsequently passed to the local control level which has a finer time scale as a reference signal. The primary goal of this study is to design an operational anticipatory control used for strategic level decision support in daily repeated traffic operations. We concern ourselves with steady-state route choice response, i.e. equilibrium flow patterns.

The objective we consider here is optimizing total travel time for the real system; however, any global objective (including environmental goals) can in principle be considered. A main bottleneck with respect to real operations is the presence of uncertainty in the model used to anticipate the route equilibrium response. In general, the real route choice response can never be precisely described and it is usually approximated by equilibrium flow models, while empirical evidence shows that only part of the users’ choices actually follow optimal route choice principles (Vreeswijk et al., 2014). In the presence of model uncertainty, due to e.g. structural model mismatch, inaccurate parameters, measurement noise, and some other unexpected disturbances, implementing control settings in real traffic networks could result in suboptimal traffic operations, like for instance unpredicted traffic congestion and spillback. A natural way to deal with model uncertainty is to make use of feedback information derived from measurement data. Using feedback mechanisms to compensate for model uncertainty has been extensively investigated for traffic applications (Hegyi et al., 2005; Dinopoulou et al., 2006; Aboudolas et al., 2009; Lin, 2011). The existing studies mainly concern on within-day traffic evolution, whereas in this work we intend to explore a new feature of the feedback mechanism applying along a day-to-day dimension within an adaptive control scheme.

Since traffic operates repeatedly from day to day, different measurements on daily traffic states such as flow or density are available. An iterative two-step method is usually adopted for an adaptive implementation scheme (Bellemans et al., 2002), repeatedly estimating model parameters by using measurements and using the updated model to generate new control settings via optimization. As suggested by Wang et al. (2006), a real-time motorway network traffic surveillance tool called RENAISSANCE that enables traffic state estimation based on real-time measurements can provide more accurate information and may help traffic operators to make more confident control decisions to improve traffic operating conditions. However, simple iterating between successive solutions of an estimation and an optimization problem usually fail to converge to the true optimal condition (Roberts and Williams, 1981), especially regarding the structural model mismatch that is inevitable as models employed for optimization are only imperfect abstractions of real system behavior. In this
context, two issues are under concern: whether the iterations could converge, and if so, whether the convergent point coincides with the real optimum.

To overcome the limitation of a traditional two-step approach, this study aims at designing an adaptive anticipatory control that is capable of generating a sequence of control settings able to converge to the true optimal operation despite model uncertainty. This requirement on control design motivates the utilization of a novel Iterative Learning Control (ILC) technique, which has been widely adopted in robot manipulators with a similar concern for robotic control design (Arimoto et al., 1984; Bristow et al., 2006). Indeed, for the repeated traffic operations, there is a high potential to learn the deviations of reality from model expectations, hence to make better control strategies. ILC works on repeatedly operating systems and it is capable of using measurements to compensate for uncertainty along the iteration domain. This property of iteration-domain feedback enables ILC to work well in devising an adaptive traffic signal scheme. An iterative learning approach was already proposed for the adaptive traffic anticipatory control in a previous study (Huang et al., 2014), modifying daily signal settings such that the generated control scheme keeps track of the real flow patterns and yields convergence on real traffic operations. The underlying mechanism of the iterative learning approach on tracking the real flow response is an implicit correction on a model bias, defined as the difference between measured and predicted flows. The modified model bias is used to generate a new signal setting in a way that the deviations are compensated for iteratively. However, the optimality of the convergent point is not guaranteed in this implicit method.

This new paper adds to our previous work the improvement in solution optimality. It follows the idea of iterative optimizing control on a basis of model bias. However, instead of implicitly updating model bias in the constraint of the optimization problem, this paper extends the method by performing an explicit first-order correction on the model bias. Flow sensitivity (Jacobian) is calculated to obtain the derivatives of both model and real flows with respect to signal settings. The derivative information is then used to formulate the first-order approximation on model bias correction, proposing an iterative optimizing control method with explicit model bias correction. The advantage over an implicit method is that the true optimum can be achieved because of the Jacobian modification; moreover the explicit correction could also be used for other purpose than optimal signal control (e.g. information provision).

1.1 Contributions of the paper

In conclusion, the main contributions of the presented paper are summarized in the following points:

1. In this paper, we formulate an adaptive anticipatory traffic control problem in the context of inaccurate network equilibrium modeling. The main idea is to apply feedback mechanism along a day-to-day dimension and hence design a control scheme used as a decision support for daily repeated traffic operations.

2. We propose an iterative optimizing algorithm with two desirable properties: providing convergence on the real traffic system, and providing consistence between the convergent point and the true optimal operating condition.

3. A first-order model bias correction is performed for the proposed algorithm, correcting both the value and curvature of the model bias. A reality-tracking metamodel is thereby defined, which is then
utilized by the control optimization procedure. This enables the traffic managers to make more confident control decisions, meanwhile, to improve the underlying decision support iteratively.

4. We identify and prove that a crucial role in improving the solution optimality is played by Jacobian of the (equilibrium) flows with respect to signal control variables. Furthermore, regarding traffic operations in reality, we present an operational sensitivity analysis and introduce a measurement-based finite difference method to estimate the operational Jacobian of the real flow response with respect to signal settings.

1.2 Outline of the paper

The remainder of the paper is organized as follows. The model uncertainty problem faced in designing an adaptive anticipatory control is introduced and an iterative optimization problem is formulated in section 2. We focus on an unknown but deterministic uncertainty due to structural modeling error. Section 2 also introduces an iterative learning approach for an adaptive traffic control strategy. In section 3, the Iterative Optimizing Control with Model Bias Correction (IOCMBC) algorithm is proposed and elaborated. A Necessary Optimality Condition (NOC) is discussed ensuring that the optimal solution to the IOCMBC is consistent with the true optimum. Furthermore, the local convergence properties are analyzed, giving some insights into sufficient conditions for convergence of IOCMBC. Section 4 discusses a crucial issue regarding requirement on the Jacobian information of the real system. A measurement-based method is implemented for real Jacobian estimation in this study in order to avoid an impractical way of performing additional perturbations. The proposed IOCMBC is tested in section 5 in a toy traffic network, compared with a traditional two-step method of iterative parameter estimation and control optimization. Section 6 briefly discusses assumptions of full observability and absence of noise on link flow measurements, and relaxation of the assumptions for large-scale applications. Section 7 shows the conclusions and discussions on future study.

2 Iterative optimization for adaptive anticipatory control

This section explains some fundamental issues regarding the adaptive anticipatory control and a design method using iterative learning approach.

2.1 Adaptive anticipatory control dealing with anticipatory equilibrium flow

Anticipatory control optimizes traffic signal timings taking into account that a modification in traffic control could change flow patterns as a result of route choice response (Taale and Van Zuylen, 2003; Taale, 2008). Several formulations can be used for modeling a traffic assignment procedure. In this paper, we describe a mutually consistent system, in that the flow patterns determine travel costs including signal delays, while in turn travel costs decide flow patterns through their impact on route choice behavior. A static network loading, more specific (and without loss of generality) a linearized BPR cost function is used to represent traffic supply. As for traffic demand model, a stochastic route choice principle is adopted, which is formulated as a discrete choice model derived from random utility theory (Cascetta, 2009).
\[ c = C(f, g) = c^0 + A \frac{f}{g_s} \]  
\[ f = F(c) = dBp(-B^Tc) \] (2.1)

Here, \( c \) is the vector of link costs, which is a function of link flows \( f \) and signal settings \( g \) (we refer to signal green splits here, thus omitting the cycle length). In the cost function \( C(.) \), \( c^0 \) is the free-flow costs and \( s \) the vector of link saturation flows, while \( A \) is the coefficient matrix in the cost function; in this study, \( A \) is a diagonal matrix whose entries can be regarded as different \( \alpha \) values (in a BPR cost function) for each link. In the route choice model \( F(.) \), \( d \) denotes the traffic demand and \( B \) the link-route incidence matrix. Route choice probabilities are denoted by \( \rho \) depending on link costs.

In this study, we focus on the steady-state control and the solution to a fixed-point model is used to describe the network equilibrium, assuming that this solution is unique. As shown in the following equation, the equilibrium solution \( f^{Eq} \) depends on a given green split \( g^0 \) (Cantarella et al., 2012).

\[ f^{Eq} = F(C(f^{Eq}, g = g^0)) \]

In the context of daily traffic operations, it is important to distinguish the real traffic system and the model. Let the equilibrium flow of reality \( f^{mea} \in \mathbb{R}^{n_f} \) for \( n_f \) links be represented by a mapping \( f^{real} : \mathbb{R}^{n_f} \rightarrow \mathbb{R}^{n_f} \). The superscript \( real \) means a perfect mathematical description of reality, while \( mea \) indicates that this result of real flow response is usually measured from traffic networks.

\[ f^{mea} = f^{real}(g) \] (2.2)

Full observability and absence of noise are assumed for the link flow measurements in this study. We assume that all the link flows are observable and noise-free, as the objective is to develop and verify a new adaptive traffic control strategy. Applications in large-scale networks in which measurement noise and missing data are detrimental to the quality of the control strategy will be discussed in section 6 of this paper.

The steady-state anticipatory control optimization problem for reality is formulated as follows.

\[
\begin{align*}
\min_{g} \quad & z(g, f^{mea}) \quad \text{(2.3a)} \\
\text{s.t.} \quad & f^{mea} = f^{real}(g) \quad \text{(2.3b)} \\
& f^{mea} \geq 0 \quad \text{(2.3c)} \\
& g^L \leq g \leq g^U \quad \text{(2.3d)}
\end{align*}
\]

Where: \( z : \mathbb{R}^{n_f} \times \mathbb{R}^{n_f} \rightarrow \mathbb{R} \) is the objective function for control optimization, e.g. total travel costs. In this paper, we assume that this function is known and can be evaluated from measurements. Equation (2.3b) is the flow prediction denoting that an equilibrium flow pattern is calculated based on signal settings. Formula (2.3c) is a non-negativity constraint on the mathematical modeling of the flows. (2.3d) restricts the boundary for the signal settings.
In general, a perfect mathematical description of reality is rarely available for control design and an approximated model is usually adopted:

\[ f^{Eq} = f(g, \mu) \quad (2.4) \]

With \( f : \mathbb{R}^n_x \times \mathbb{R}^n_{\mu} \rightarrow \mathbb{R}^n_y \) representing the equilibrium flow model, and \( \mu \in \mathbb{R}^n_{\mu} \) denoting a set of model parameters which is adjustable for modeling accuracy. In this paper we are concerned about demand side model uncertainty thus we assume that the link cost function is accurately known.

Signal settings in the model and reality are the same. Two sources of model uncertainty are recognized for (2.3), one being inaccurate values for \( \mu \) and the other a structural model-reality mismatch \( f(.) \neq f^{real}(.) \). Using such model approximation, solution to the original problem (2.3) can be approximated by solving the following model-based optimization problem.

\[
\begin{align*}
\min_{g} & \quad z(g, f^{Eq}) \\
\text{s.t.} & \quad f^{Eq} = f(g, \mu) \\
& \quad f^{Eq} \geq 0 \\
& \quad g^L \leq g \leq g^U
\end{align*}
\]

First, we introduce a feasible solution mapping with respect to the equilibrium flow for this optimization problem (2.5):

\[ g := G(f^{Eq}) := \{ g \in [g^L, g^U] : f(g, \mu) \geq 0 \} \quad (2.6) \]

An Anticipatory Equilibrium Flow (AEF) is defined as the flow pattern associated with the solution to the optimization problem (2.5). We call it a model-based AEF denoted as \( f^{Eq}* \). The corresponding optimal signal setting can be written as:

\[ g^* = G^*(f^{Eq}*) := \{ g \in G(f^{Eq}) : z(g, f^{Eq}) = z^* \} \quad (2.7) \]

Similarly, a true AEF is defined for the global optimization problem (2.3) denoted as \( f^{mea}* \) and the corresponding true optimal signal setting is denoted as \( g^{true}* \). Note that \( f^{mea}* \) is different from the measured equilibrium flow \( f^{mea} \); it relates to the true optimal performance in real traffic networks.

In this study, the concern for an adaptive anticipatory control is to iteratively adjust model-based AEF \( f^{Eq}* \) in order to track true AEF \( f^{mea}* \) with the help of measurements.

### 2.2 An iterative learning approach

Implementing signal settings calculated from problem (2.5) in reality usually results in suboptimal operations resulting in even some unexpected traffic congestions due to model-reality mismatch. Regarding the fact that traffic operates repeatedly e.g. from day to day and measurements on traffic flows are available for handling uncertainty, iterative optimizing control methods can be utilized for designing adaptive anticipatory control strategies. In this context, and for the sake of illustration, one
iteration is regarded as a day, but one may as well consider periods of weeks or months of operation before the model is corrected and the signal is updated. This interpretation follows a control setting procedure that flow observed today (more general: of the previous period) is used for updating new signal settings for tomorrow (more general: for the next period). A two-step implementation scheme is conventionally applied. The idea is to iteratively estimate selected model parameters and generate new signal settings via control optimization. A possible two-step scheme can be written using the following two equations representing the two problems of parameter estimation and control optimization:

\[
\mu_k = \arg \min_{\mu} \| f^{mea}_k - f(g_k, \mu) \| \quad (2.8a)
\]

\[
g_{k+1} = \arg \min_{g} z(g, f(g, \mu_k)) \quad (2.8b)
\]

The constraints are neglected for simplicity. Formulation (2.8a) shows one instance of minimizing the sum of square errors between the measured and modeled flows for parameter estimation. However we do not exclude other estimation techniques. The parameter estimation procedure uses measurements obtained from the current kth iteration control setting to correct the parameters; and then the control optimization step determines a new signal after having fixed these estimated parameters. The two problems interact and several iterations may be required until no further improvement is obtained. Similar to comments on using an iterative optimization assignment (IOA) approach to solve an interaction between traffic assignment and traffic control (Allsop, 1974; Gartner, 1975), it is well known that simple iterating between two steps is not guaranteed to converge even to a local optimum (Smith, 1979; Dickson, 1981). Indeed, as the estimation is performed with a previously applied signal setting, it might be unrelated to the next control optimization. Therefore, minimizing the square error in flow may not help in optimizing the total travel time. Figure 1 shows a symbolic representation of a traditional two-step procedure. An optimal signal \( g_k^* \) for the kth iteration is solved based on the kth estimated model parameters; then it is used for a (k+1)th estimation which is then used for solving \( g_{k+1}^* \). A simple two-step approach will not converge to the true optimum on the real system unless the model is structurally correct and the parameters are identifiable. However, in the presence of model mismatch, which is an inherent attribute in modeling, two questions immediately arise: 1) if the control settings generated based on an inaccurate model can yield a convergence on real traffic operations? and 2) if the convergent solution matches the true optimal performance?
In a previous study (Huang et al., 2014), an iterative learning approach has been proposed for handling the model mismatch. This iterative learning method is constructed with an important property of reality-tracking, i.e. the generated control settings keep track of the true flow patterns and yield convergence on real traffic operations. It is motivated by an Iterative Learning Control (ILC) technique, which has been widely applied for industrial applications on robotic manipulator as well as chemical process control. Many rule-based control laws have been elaborated for regulating the controlled system to a reference state (Park et al., 1999; Moore, 2001; Chen and Moore, 2002; Hou et al., 2008; Huang et al., 2013). This is followed by developments on optimization-based methods, in order to have a more systematic design concept and enable more implications on real applications (Lee et al., 2000; Gunnarsson and Norrlöf, 2001; Owens and Hätönen, 2005). A similar iterative trial-and-error implementation scheme has also been developed for traffic tolling on road networks in the absence of demand functions (Yang et al., 2004; Han et al., 2004).

An adaptive anticipatory control is formulated following an optimization-based ILC design paradigm. The problem can be interpreted as the determination of the (k+1)th control setting as an input to the traffic system that can reduce the total travel cost in an optimal way, whilst adding penalties to account for the model uncertainty.
\[
\min_{\mathbf{g}_{k+1}} \omega.f(\mathbf{g}_{k+1}, \mathbf{f}_{k+1}) + \mathbf{g}_{k+1}^T \mathbf{W}_g \mathbf{g}_{k+1} + \Delta \mathbf{g}_{k+1}^T \mathbf{W}_\Delta \Delta \mathbf{g}_{k+1} \quad (2.9a)
\]

\[
\begin{aligned}
\text{s.t. } & \mathbf{f}_{k+1} = f_{k+1}^{mea} + (f(\mathbf{g}_{k+1}, \mathbf{f}) - f_{k+1}^{Eq}) \\
& \Delta \mathbf{g}_{k+1} = \mathbf{g}_{k+1} - \mathbf{g}_k \\
& \mathbf{g}^L \leq \mathbf{g}_{k+1} \leq \mathbf{g}^U \quad (2.9b, c, d, e)
\end{aligned}
\]

The use of measurements for handling uncertainty is mainly reflected by (2.9b). It indicates that a prediction on flow response \( \mathbf{f}_{k+1} \), which is a modification of an observation upon the completion of the \( k \)th iteration, is used in the optimization procedure to calculate a new control setting \( \mathbf{g}_{k+1} \). It should be generated in a way that, under the basic model information, its deviation to a model-reality mismatch from the last iteration. Formula (2.9c) is the non-negativity constraint on the flow. Equation (2.9d) shows the signal update while (2.9e) indicates that the signal green time is bounded by minimal and maximal green time.

\[
\omega \in \mathbb{R}, \mathbf{W}_g \in \mathbb{R}^{n_g \times n_g}, \text{ and } \mathbf{W}_\Delta \in \mathbb{R}^{n_\Delta \times n_\Delta} \text{ are weights on total travel cost performance, signal inputs and signal updates respectively. They are design parameters which can be tuned to balance the adaptation of signal and the convergence of the algorithm.}
\]

The objective for this iterative optimization problem (2.9) is to design a sequence of anticipatory control settings that is capable of achieving a true optimal performance associated with the real traffic operations. Two main arguments are here derived.

1) In a traditional two-step approach, the selected model parameters are updated by using measurements. While the proposed ILC-based method keeps the parameters at some default values (for a nominal model). This is justifiable for the situations under which model calibration for a better description on real system is not a main concern and the ultimate and only goal is to design control settings that can operate well in reality.

2) By introducing a concept of model bias \( \mathbf{b}_k \in \mathbb{R}^{n_f} \) defined as difference between measured and model predicted flow shown in (2.10), it can be observed that the main idea behind the ILC-based method is actually to manipulate the model bias for tracking reality. The model bias is updated at each iteration implicitly as a constraint to the optimization problem:

\[
\mathbf{b}_k = f_{k+1}^{mea} - f_{k+1}^{Eq} = f^\text{real}(\mathbf{g}_k) - f(\mathbf{g}_k, \mathbf{f}) \quad (2.10)
\]

As discussed, in the two-step approach, parameter estimation is addressed with a current control setting while control optimization is performed with a determined parameter value. The reason of its failure actually lies in that the objective of parameter estimation might be unrelated to the objective of performance optimization. Model bias plays a significant role in connecting the objectives of estimation and optimization. Some approaches to integrate these objectives have been explored in literature applied in the real-time optimization problems (Cutler and Perry, 1983; Chachuat et al., 2009). Adaptations are usually performed on the objective function and/or the constraint functions (Roberts and Williams, 1981; Roberts, 1995; Srinivasan and Bonvin, 2002; Marchetti et al., 2007). Our proposed method follows the integration idea; while differently from the previous approaches, we address the adaptation on the system model bias.
In order to present the role of model bias, let us reformulate (2.9a) as (2.11):

\[ f_{k+1} - f(g_{k+1}, \mu) = b_k \]  

(2.11)

Equation (2.11) indicates that model bias update introduces a one-step prediction on the deviations of model from reality. Hence, the impact of the subsequent signal settings is counted in modifying the deviations. In this sense, model bias is in essence a connection between tracking reality and optimizing control. It is worth noting that in this approach, adaptation on model bias instead of model parameters is performed for tracking reality, as well as for supporting the subsequent control decision-making. Default values are applied for the model parameters.

The ILC-based method is a direct control method and compensates for the model uncertainty via implicit model bias update together with penalties in the objective function. It suggests an effective way of determining optimal control capable of tracking real operating conditions and yields a satisfactory practical application. However, the implicit model bias update therein is actually a partial correction in a sense that only the value of the modeled flow is adjusted based on the measurements, leaving the sensitivity of the model unchanged. Therefore, penalties on the control settings are added for regulating a reality-tracking convergence. As a result, there is a trade-off on the optimal total travel cost regarding a convergent scheme in the presence of model uncertainty; the convergent point might be suboptimal and does not match the true optimum. Further efforts are needed if an improvement in optimality is required. The present paper works on matching with the true optimum. A Jacobian modification term is added to the model bias prediction and hence a full correction on model bias is obtained. The model bias correction is explicitly performed and used for a subsequent total travel cost optimization which is separated from the penalized optimization problem. The approach proposed in this study is to construct and describe in detail an algorithm with a desired property of, in addition to reality-tracking, matching of the convergent solution with the true optimum. The following section 3 elaborates on design of an adaptive anticipatory control strategy using iterative control optimization considering explicit model bias correction.

3 Iterative control optimization with explicit model bias correction

3.1 Model bias correction

As discussed in section 2, in the presence of model mismatch, traffic operations might deteriorate due to the inconsistency between the model-based optimum and the true optimum. A full and explicit model bias correction is proposed in this section to compensate for the uncertainty. The basic idea is to use measurements to adjust both the value of the model output and the gradient of the model, which captures the sensitivity of the output to the input changes. More specifically, a first-order prediction is performed on the model bias, updating both the value and the gradient. Following this way, we intend to reduce the numerical error generated from approximation by explicitly adding sensitivity information to the model bias correction. As the model bias is determined from a complete set of link flow observations, the model bias correction is constructed based on the assumption of full observability on link flows. A brief discussion on how to relax this assumption is arranged in section 6.
Standing at a current k\textsuperscript{th} iteration, a flow pattern \( f_{k+1} \) denotes a prediction on real flow patterns when the decision \( g_{k+1} \) for the next (k+1)\textsuperscript{th} iteration is implemented; thus \( f_{k+1} \) depends on \( g_{k+1} \). An expected model bias \( b_{k+1} \in \mathbb{R}^{n_f} \) is defined as its difference from a model prediction. The value of model bias at k\textsuperscript{th} iteration is observed as the distance between measured and predicted flow patterns. Curvature of the bias is captured by the Jacobian around the current operating point. Therefore a model bias correction is performed as follows:

\[
\begin{align*}
\hat{b}_{k+1} &= b_k + \left( \frac{\partial b}{\partial g} \right)_{|r_k} (g_{k+1} - g_k) \\
&= f_{k+1} - f(g_{k+1}, \mu) \\
&= f_{\text{real}}(g_k) - f(g_k, \mu) + \delta_k (g_{k+1} - g_k)
\end{align*}
\tag{3.1}
\]

Note that \( b_k \) is an observation from reality as shown in (2.8), whereas \( b_{k+1} \) is a prediction on model bias for the next iteration:

\[
\begin{align*}
\hat{b}_{k+1} &= f_{k+1} - f(g_{k+1}, \mu) \\
&= f_{\text{real}}(g_k) - f(g_k, \mu) + \delta_k (g_{k+1} - g_k) \\
&= f_{k+1} - f(g_{k+1}, \mu)
\end{align*}
\tag{3.2}
\]

Substituting equation (2.10) with (3.1) and (3.2):

\[
\begin{align*}
f_{k+1} - f(g_{k+1}, \mu) &= f_{\text{real}}(g_k) - f(g_k, \mu) + \delta_k (g_{k+1} - g_k) \\
&= \frac{\partial f_{\text{real}}}{\partial g} \left|_{r_k} \right. - \frac{\partial f}{\partial g} \left|_{r_k} \right.
\end{align*}
\tag{3.3}
\]

in which, \( \delta_k \in \mathbb{R}^{n_x \times n_f} \) denotes the Jacobian error between reality and model \( \delta_k = \frac{\partial f_{\text{real}}}{\partial g} \left|_{r_k} \right. - \frac{\partial f}{\partial g} \left|_{r_k} \right. 
\]

Therefore, the expectation flow for the next control optimization, which is adjusted on the basis of model bias correction, is written as:

\[
\begin{align*}
f_{k+1} = f(g_{k+1}, \mu) + b_{k+1} - f(g_{k+1}, \mu) + b_k + \delta_k (g_{k+1} - g_k) \\
&= f(g_{k+1}, \mu) + \hat{b}_{k+1} + \delta_k (g_{k+1} - g_k)
\end{align*}
\tag{3.4}
\]

This brings a reality-tracking metamodel \( f_{\text{meta}}^{k} : \mathbb{R}^{n_f} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_f} \), which is associated with real operations. Metamodels are utilized by the optimization procedure to calculate a new optimal signal setting (Osorio and Bierlaire, 2013). We will refer to model (3.5) as ‘metamodel’ for short in the rest of the paper:

\[
\begin{align*}
f_{\text{meta}}(g, \mu) &= f(g, \mu) + b_k + \delta_k (g - g_0)
\end{align*}
\tag{3.5}
\]

Figure 2 shows a symbolic representation of the correction on model bias along one signal input direction. Observation \( b \) is the value of the model bias denoting distance between the measured and the modeled flows, whereas \( \delta \) corresponds to the gradient of the model bias denoting slope error between the sensitivity of the real and the modeled flow to the signal changes. Definition (3.5) tells that a (k+1)\textsuperscript{th} prediction on real flow i.e. the metamodel, is derived by adjusting an underlying system model with a model bias correction on both distance and slope. As shown in figure 2, upon the completion of the k\textsuperscript{th} iteration, the system model is offset with an observed distance \( b_{\text{opt}} \), meanwhile rotated with a slope error \( \delta_{\text{opt}} \) estimated in the k\textsuperscript{th} iteration. Note that the nominal system model itself does not change along iterations. A critical manipulation during the model bias correction and metamodel adaption is calculating gradients of the real flow response. This key issue of real gradient (or Jacobian if control \( g \) has multiple dimensions) calculation will be elaborated in the next section.
3.2 The IOCMBC method

Taking into account an explicit and full model bias correction, an algorithm called Iterative Optimizing Control with Model Bias Correction (IOCMBC) is proposed for an adaptive anticipatory control scheme. By explicitly performing a full model bias correction, the adaptive scheme actually follows a combined two-stage framework of model bias correction and control setting update. The metamodel (3.5), which is modified based on the model bias correction serves as a basis for control optimization. Hence, one advantage of this two-stage procedure is that it enables the traffic managers to improve the underlying control decision support system, through improving the reality-tracking metamodel. Figure 3 shows the updating procedure of the proposed control scheme.
An optimal signal setting $g_{k+1}^*$ for the $(k+1)^{th}$ iteration is determined by solving the following model-based optimization problem:

$$
g_{k+1}^* = \arg \min_g z(g, f_{meta}(g, \mu)) \quad (3.6a)
$$

s.t. $f_{meta}(g, \mu) = f(g, \mu) + b_k + \delta_k (g - g_k)$ \quad (3.6b)

$$
f_{meta}(g, \mu) \geq 0 \quad (3.6c)
$$

$$
g^L \leq g \leq g^U \quad (3.6d)
$$

A prediction on the flow response that is used in the objective function results from the metamodel. Equation (3.6b) describes that the metamodel is a modification on model prediction with a full first-order model bias correction. Non-negativity and boundary constraints are included in (3.6c) and (3.6d) respectively.

Considering that even $g_{k+1}^*$ as a solution to (3.6) will not be optimal at once due to omitting higher-order terms in the bias correction, it may be cautious to consider the newly obtained $g_{k+1}^*$ as an optimization direction for the next iterate, rather than the next control value to implement for the next iteration. We then need to determine the step size into this optimization direction to find new implementation signal setting $g_{k+1}$. Naturally, an exponential smoothing structure is adopted:

$$
g_{k+1} = (I - K)g_k + Kg_{k+1}^* \quad (3.7)
$$

Here $K \in \mathbb{R}^{n_g \times n_g}$ is a gain matrix representing a suitable step from $g_k$ to $g_{k+1}^*$ and usually takes a value of $K = \text{diag}(\lambda_1, \ldots, \lambda_{n_g})$, in which $\lambda_1, \ldots, \lambda_{n_g} \in [0,1]$, hence allowing in principle different step sizes for the different dimensions of $g$. In many cases, $K$ is regarded as a design parameter and serves as a practical setting for regulating convergence. Note that when one chooses $K$ equal to $I$, no smoothing, hence full step size is considered.
The new control \( g_{k+1} \) is then implemented in reality. Model bias is updated, including the observed value at the current iteration and the Jacobian that is also estimated around the current operating point:

\[
b_{k+1} = f^{\text{real}}(g_{k+1}) - f(g_{k+1}, \mu) \quad (3.8a)
\]

\[
\delta_{k+1} = \left. \frac{\partial f^{\text{real}}}{\partial g} \right|_{g_{k+1}} - \left. \frac{\partial f}{\partial g} \right|_{g_{k+1}} \quad (3.8b)
\]

The IOCMBC algorithm is now summarized as follows.

- **Step 1:** Set a default value for model parameter \( \mu \); choose design parameters \( K \); set initial signal setting \( g_0 \).
- **Step 2:** Apply the initial signal setting; implement \( g_0 \) in reality, calculate the initial \( f_0^{\text{Eq}} \) and measure \( f_0^{\text{mea}} \). Then solve \( g_1^* \) from the control optimization problem, set a step from \( g_0 \) to \( g_1^* \) and derive \( g_1 \), set \( k=1 \);
- **Step 3:** Apply signal settings; implement \( g_k \) in reality, calculate mode prediction equilibrium flow \( f_k^{\text{Eq}} \) and measure the resulting flow \( f_k^{\text{mea}} \);
- **Step 4:** Model bias correction; update model bias \( b_k = f^{\text{real}}(g_k) - f(g_k, \mu) = f_k^{\text{mea}} - f_k^{\text{Eq}} \), estimate Jacobian for both model and reality at the current operating point and derive the Jacobian error \( \delta_k = \left. \frac{\partial f^{\text{real}}}{\partial g} \right|_{g_k} - \left. \frac{\partial f}{\partial g} \right|_{g_k} \);
- **Step 5:** Update reality-tracking metamodel; update the metamodel on the basis of model bias correction and derive a prediction flow for updating the next signal control;

\[
f_{k+1} = f^{\text{meta}}(g_{k+1}, \mu) = f(g_{k+1}, \mu) + b_k + \delta_k (g_{k+1} - g_k)
\]

- **Step 6:** Find optimization direction; calculate \( g_{k+1}^* \) by solving the control optimization for the updated metamodel, and derive an optimization direction \( (g_{k+1}^* - g_k) \);
- **Step 7:** Set appropriate step size and update signal control: \( g_{k+1} = (I - K)g_k + Kg_{k+1}^* \);
- **Step 8:** Check termination; stop if the termination condition is satisfied, otherwise set \( k=k+1 \) and go to step 3.

### 3.3 Matching of necessary optimality conditions

The goal of this study is to design an adaptive control scheme in favor of real operations, whereby the optimal solution from the iterative optimization problem should match the true optimal operating point. IOCMBC can bring up this benefit. Upon convergence, the resulting optimal solution is also the solution to the true optimization problem. This property is a prerequisite for iteratively solving the optimization problem (2.5) to approximate the true solution to the problem (2.3).
To show this capability of IOCMBC, first the Necessary Optimality Conditions (NOC) for a general optimization problem is introduced. Recall the true optimization problem (2.3) in section 2, suppose that \( \mathbf{g}^{\text{true}} \) is a local minimum of (2.3) and that the required constraint qualification holds at \( \mathbf{g}^{\text{true}} \) (Boyd and Vandenberghe, 2004), and the function \( z(.) \) and \( f(.) \) are differentiable at \( \mathbf{g}^{\text{true}} \). Then we introduce the Lagrange multiplier vectors \( \mathbf{l}_f \in \mathbb{R}^{p_f}, \mathbf{l}_g \in \mathbb{R}^{p_g}, \mathbf{l}_{g^L} \in \mathbb{R}^{p_{g^L}} \) such that the following conditions are satisfied at \((\mathbf{g}^{\text{true}}, \mathbf{l}_f, \mathbf{l}_{g^L}, \mathbf{l}_{g^U})\):

\[
\nabla_{\mathbf{g}} L(\mathbf{g}^{\text{true}}, \mathbf{l}_f, \mathbf{l}_{g^L}, \mathbf{l}_{g^U}) = 0
\]

\[
f^{\text{real}}(\mathbf{g}^{\text{true}}) \geq 0
\]

\[
\mathbf{g}^L \leq \mathbf{g}^{\text{true}} \leq \mathbf{g}^U
\]

\[
l_f \geq 0, l_{g^L} \geq 0, l_{g^U} \geq 0
\]

\[
l_f^T f^{\text{real}}(\mathbf{g}^{\text{true}}) = 0, l_{g^L}^T (\mathbf{g}^{\text{true}} - \mathbf{g}^L) = 0, l_{g^U}^T (\mathbf{g}^U - \mathbf{g}^{\text{true}}) = 0
\]

Where \( L(\mathbf{g}^{\text{true}}, \mathbf{l}_f, \mathbf{l}_{g^L}, \mathbf{l}_{g^U}) \) is the Lagrangian function and taking the derivatives as:

\[
\nabla_{\mathbf{g}} L(\mathbf{g}^{\text{true}}, \mathbf{l}_f, \mathbf{l}_{g^L}, \mathbf{l}_{g^U}) = \left[ \frac{\partial z}{\partial \mathbf{g}} \right]_{\mathbf{g}^{\text{true}}=\mathbf{g}^{\text{true}}_k} + \left[ \frac{\partial f^{\text{real}}}{\partial \mathbf{g}} \right]_{\mathbf{g}^{\text{true}}=\mathbf{g}^{\text{true}}_k} - \left[ \frac{\partial f^{\text{real}}}{\partial \mathbf{g}} \right]_{\mathbf{g}^{\text{true}}=\mathbf{g}^{\text{true}}_k} l_f - l_{g^L} + l_{g^U}
\]

The first-order NOC conditions are often known as Karush-Kuhn-Tucker (KKT) conditions (Boyd and Vandenberghe, 2004). We refer to the NOC point for problem (2.3) as a true NOC point.

Similarly, a model-based NOC point is also defined at \( \mathbf{g}^* \) based on the KKT conditions for the model-based optimization problem (2.5), replacing \( f^{\text{real}}(\mathbf{g}^{\text{true}}) \) with \( f(\mathbf{g}^*, \mathbf{\mu}) \).

Based on the true NOC and model-based NOC, together with the definitions of true AEF and model-based AEF mentioned introduced in section 2, we present an important feature of IOCMBC. That is, by implementing the proposed IOCMBC algorithm, the resulting AEF state for the model-based optimization problem (3.6) matches the true AEF state, and the resulting NOC point is also a true NOC point. This feature is formalized in the following Proposition 1.

**Proposition 1 (AEF and NOC matching).** Suppose that the gain matrix \( \mathbf{K} \) is nonsingular and the IOCMBC algorithm described by (3.6), (3.7) and (3.8) converges to a metamodel-based AEF state \( \mathbf{f}^{\text{meta}} \); the convergent point \( \mathbf{g}_k^* = \lim_{k \to \infty} \mathbf{g}_k \) is a NOC point for the problem (3.6). Then convergent implementation point \( \mathbf{g}_\infty = \lim_{k \to \infty} \mathbf{g}_k \) coincides with the true NOC point for the traffic control optimization problem (2.3), and \( \mathbf{f}^{\text{meta}} \) coincides with the real optimal traffic state \( \mathbf{f}^{\text{mea}} \).

**Proof.** Let \( k \to \infty \), from (3.7) we have:

\[
\mathbf{g}_\infty = \mathbf{g}_k^*
\]

After some manipulations on (3.6b), it can be shown that
\[ f_{\text{meta}}(g_\infty^*, \mu) = f(g_\infty^*, \mu) + f_{\text{real}}(g_\infty^*) - f(g_\infty^*, \mu) \]  
(3.11)

That is
\[ f_{\text{meta}}(g_\infty^*, \mu) = f_{\text{real}}(g_\infty^*) \]  
(3.12)

Calculating the derivatives of \( f_{\text{meta}}(g, \mu) \) with respect to \( g \) around \( g_\infty^* \)

\[ \frac{\partial f_{\text{meta}}}{\partial g} \bigg|_{g_\infty^*} = \frac{\partial f}{\partial g} \bigg|_{g_\infty^*} + \left( \frac{\partial f_{\text{real}}}{\partial g} \bigg|_{g_\infty^*} - \frac{\partial f}{\partial g} \bigg|_{g_\infty^*} \right) \]  
(3.13)

We obtain
\[ \frac{\partial f_{\text{meta}}}{\partial g} \bigg|_{g_\infty^*} = \frac{\partial f_{\text{real}}}{\partial g} \bigg|_{g_\infty^*} \]  
(3.14)

By assumption \( g_\infty^* \) is a NOC point for the problem (3.6), it is obvious that it satisfies the following conditions with the specified Lagrange multipliers \( l_f, l_g^L, l_g^U \)

\[ \nabla_g L_{\text{meta}}(g_\infty^*, l_f, l_g^L, l_g^U) = 0 \]  
(3.15a)

\[ f_{\text{meta}}(g_\infty^*, \mu) \geq 0 \]  
(3.15b)

\[ g^L \leq g_\infty^* \leq g^U \]  
(3.15c)

\[ l_f \geq 0, l_g^L \geq 0, l_g^U \geq 0 \]  
(3.15d)

\[ l_f^T f_{\text{meta}}(g_\infty^*, \mu) = 0, l_g^T (g_\infty^* - g^L) = 0, l_g^U (g^U - g_\infty^*) = 0 \]  
(3.15e)

In which
\[ \nabla_g L_{\text{meta}}(g_\infty^*, l_f, l_g^L, l_g^U) = \frac{\partial f_{\text{meta}}}{\partial g} \bigg|_{g_\infty^*} + \frac{\partial \mu}{\partial g} \bigg|_{g_\infty^*} - \frac{\partial f_{\text{meta}}}{\partial g} \bigg|_{g_\infty^*} l_f - l_g^L + l_g^U \]

From (3.10), (3.12) and (3.14), \( g_\infty^* \) satisfies (3.15) and is consistent with a true NOC point \( g^{\text{true}} \) associated with the original problem (2.3).

Besides, the true AEF state \( f^{\text{true}} \) is represented as
\[ f^{\text{true}} = f_{\text{real}}(g^{\text{true}}) \]  
(3.16)

(3.10), (3.12) and (3.16) verifies that metamodel-based AEF \( f^{\text{meta}} \) coincides with the true AEF \( f^{\text{true}} \).

□

A crucial issue for this matching property is the availability and accuracy of the Jacobian information (especially for the real flow response). The calculation of Jacobian matrices for the real system will be addressed in the next section.
Proposition 1 tells that under some conditions, a convergent solution of the IOCMBC algorithm is also a true NOC, i.e. the proposed method finds a solution to the optimization of real traffic operations. However, convergence of IOCMBC is not ensured. In the following subsection, some specifications are provided regarding the choice of the design parameters (in particular the gain matrix $K$) that could regulate the convergence. A local convergence analysis is addressed subsequently.

### 3.4 Convergence analysis

This subsection discusses convergence conditions for the proposed IOCMBC algorithm, in the absence of measurement noise. Under some specifications, applying IOCMBC can ensure leading the traffic operations towards the real optimal state.

To facilitate the convergence analysis, first a constant term for the model bias correction is introduced with a mapping $\varepsilon: \mathbb{R}^{n_x} \to \mathbb{R}^{n_f}$ showing the dependence on a current signal input $g_k$.

$$\varepsilon_k = \varepsilon(g_k) = b_k - \delta_k g_k \quad (3.17)$$

Thus the optimization problem (3.6) is rewritten as follows:

$$g_{k+1}^* = \arg \min_g z(g, f^\text{meta}(g, \mu)) \quad (3.18a)$$

s.t. $f^\text{meta}(g, \mu) = f(g, \mu) + \varepsilon_k + \delta_k g $ \quad (3.18b)

$$f^\text{meta}(g, \mu) \geq 0 \quad (3.18c)$$

$$g^f \leq g \leq g^{U} \quad (3.18d)$$

We define a solution mapping $\Omega: \mathbb{R}^{n_f} \to \mathbb{R}^{n_f}$ for solutions to this optimization problem regarding a current operation point $g_k$ such that

$$g := \Omega(\varepsilon(g_k)) := \{ g \in [g^f, g^{U}]: f^\text{meta}(g, \mu) \geq 0 \} \quad (3.19)$$

Then the optimal solution mapping is formulated with the corresponding optimal performance $z^*$.

$$g_{k+1}^* = \Omega^*(\varepsilon(g_k)) = \{ g \in \Omega(\varepsilon(g_k)) : z(g, f^\text{meta}(g, \mu)) = z^* \} \quad (3.20)$$

Regarding the convergent point of (3.6) $g_\infty^* = \lim_{k \to \infty} g_k^*$, we have

$$g_\infty^* = \Omega^*(\varepsilon(g_\infty))$$

In which $g_\infty = \lim_{k \to \infty} g_k$ is the convergent solution to IOCMBC, and $g_\infty^* = g_\infty^*$ as indicated by (3.7).

The following properties are first assumed for the studied traffic system.

**Assumption 1**: Link cost function $C(.)$ is globally Lipschitz continuous in both $f$ and $g$; link flow function $F(.)$ is globally Lipschitz continuous in $c$. 
**Assumption 2:** The sufficient optimality conditions for a local optimum of problem (3.18) at each iteration are satisfied at \( \mathbf{g}_k^* \) for \( \mathbf{g}_k^* = \Omega^* (\varepsilon(\mathbf{g}_{k-1})) \) with \( \forall k \).

**Assumption 3:** The objective function \( z(.) \) is exactly represented and is differentiable at both \( \mathbf{g} \) and \( \mathbf{f} \).

**Assumption 4:** All link flows can be measured in the traffic network under consideration and the measurements are noise-free.

Then we formalize the Proposition 2 as follows:

**Proposition 2 (local convergence of IOCMBC).** Under the assumptions 1-4, the IOCMBC algorithm described by (3.18), (3.7) and (3.8) converges if the gain matrix \( \mathbf{K} \) is chosen such that

\[
(I - \mathbf{K}) + K \left( \frac{\partial \Omega^*}{\partial \mathbf{g}_k} \right) |_{\varepsilon_k} < 1 \quad (3.21)
\]

In which \( \| \| \) represents any suitable norm.

**Proof.** It follows from the assumption 1 and 2 that there is an optimal point \( \mathbf{g}_{k+1}^* \) to the problem (3.18). According to (3.7) we formulate a fixed-point mapping for the consecutive implementation signal settings \( \Psi : \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_v} : \)

\[
\mathbf{g}_{k+1} = (I - \mathbf{K})\mathbf{g}_k + \mathbf{K}\mathbf{g}_{k+1}^* = \Psi(\mathbf{g}_k) \quad (3.22)
\]

Consider a linear approximation on \( \Psi \) around the convergent optimal point \( \mathbf{g}_\infty \)

\[
\Psi(\mathbf{g}_k) = \Psi(\mathbf{g}_\infty) + \left. \frac{\partial \Psi}{\partial \mathbf{g}_k} \right|_{\varepsilon_k} (\mathbf{g}_k - \mathbf{g}_\infty) \quad (3.23)
\]

Now we define a signal error with respect to \( \mathbf{g}_\infty \)

\[
\delta\mathbf{g}_{k+1} = \mathbf{g}_{k+1} - \mathbf{g}_\infty \quad (3.24)
\]

It is natural to derive that:

\[
\delta\mathbf{g}_{k+1} = \Psi(\mathbf{g}_k) - \Psi(\mathbf{g}_\infty) = \left. \frac{\partial \Psi}{\partial \mathbf{g}_k} \right|_{\varepsilon_k} \delta\mathbf{g}_k \quad (3.25)
\]

Following a conventional way of analyzing convergence for a linear dynamical system (e.g. Lo and Bie, 2006; Bie, 2008), we obtain the convergence conditions required for the operator norm:

\[
\left| \left. \frac{\partial \Psi}{\partial \mathbf{g}_k} \right|_{\varepsilon_k} \right| < 1 \quad (3.26)
\]

In which \( \left| \left. \frac{\partial \Psi}{\partial \mathbf{g}_k} \right|_{\varepsilon_k} \right| = (I - \mathbf{K}) + K \left( \frac{\partial \Omega^*}{\partial \mathbf{g}_k} \right) |_{\varepsilon_k} \)
Along with the convergence of the signal error \( \delta g_k = 0 \), the implementation signal setting converges to the optimal point \( g_{k+1} \rightarrow g_c \). 

Proposition 2 establishes conditions required for a local convergence of IOCMBC around the solution point \( g_c \). These conditions suggest that properly selecting the step size \( K \) can improve convergence and effectively regulate traffic system towards its true optimal operations.

### 4 Jacobian estimation

As mentioned before, a key issue for applying the proposed IOCMBC is the requirement on Jacobian information. Correction on model bias and the resulting modification on flow metamodel include Jacobian information associated with both model and reality. Challenge arises especially regarding the calculation of the Jacobian for the real flow response. This section focuses on this issue.

A most straightforward approach for calculating derivatives as discussed in literature is a finite difference approximation method (Mansour and Ellis, 2003; Gao and Engell, 2005). Basically, it consists in perturbing each input variable individually and calculating the corresponding derivative element. Implementing a finite difference method for the flow anticipation model, a component of the model Jacobian, which is a derivative of link flow with respect to signal setting, is derived:

\[
\frac{\partial f^i}{\partial g^j} = \frac{f^i(g^1...g^i + \delta g...g^n) - f^i(g)}{\delta g}
\] (4.1)

\( f^i \) is the flow for link \( i \) and \( g^j \) the signal setting for link \( j \). \( \frac{\partial f^i}{\partial g^j} \) denotes the \( ij \)th Jacobian component.

This \( ij \)th component shows the derivative of the \( i \)th flow regarding a small perturbation \( \delta g \) performed on the \( j \)th signal setting. Therefore, the Jacobian of equilibrium flow response with respect to signal control is calculated in (4.2):

\[
\frac{\partial f}{\partial g} = \begin{bmatrix}
\frac{f^1(g^1 + \delta g...g^n) - f^1(g)}{\delta g} & \ldots & \frac{f^1(g^1...g^n + \delta g) - f^1(g)}{\delta g} \\
\frac{f^n(g^1 + \delta g...g^n) - f^n(g)}{\delta g} & \ldots & \frac{f^n(g^1...g^n + \delta g) - f^n(g)}{\delta g}
\end{bmatrix}
\] (4.2)

However, it is obvious that this approach is impractical for implementation in reality as it requires performing additional perturbations on each signal setting and has to measure the flow response to the corresponding perturbation. It is necessary to wait for each equilibrium flow response in reality for obtaining all the measurements. Several methods have been proposed to avoid the additional perturbations. A so-called Broydon’s method is proposed to estimate the Jacobian for reality (Fletcher, 1980; Roberts, 2000), in which a recursive formula is defined and the output derivatives
are updated using current and past measurements. A different way of implementing a finite difference approximation is developed (Brdys and Tajewski 1994), using measurements observed in previous iterations instead of additional perturbations.

In this work, we employ a measurement-based finite difference method for estimating Jacobian for the real flow response, called operational Jacobian. The daily operational Jacobian is updated based on a measured flow set containing \((n_g+1)\) flows \(\{f_{k}^{\text{mea}}, f_{k-1}^{\text{mea}}, \ldots, f_{k-n_y}^{\text{mea}}\}\), as well as the past \((n_g+1)\) signal settings. Operational Jacobian at the current \(k\)th iteration \(\frac{\partial f^{\text{real}}}{\partial g}
\bigg|_{k}\) is calculated by the following formula (4.3):

\[
\frac{\partial f^{\text{real}}}{\partial g}
\bigg|_{k} = (\Delta g_{k})^{-1} \begin{bmatrix}
    f_{k}^{\text{mea}} - f_{k-1}^{\text{mea}} \\
    \vdots \\
    f_{k}^{\text{mea}} - f_{k-n_y}^{\text{mea}}
\end{bmatrix} = (\Delta g_{k})^{-1} \begin{bmatrix}
    f^{\text{real}}(g_{k}, \mu) - f(g_{k-1}, \mu) \\
    \vdots \\
    f^{\text{real}}(g_{k}, \mu) - f(g_{k-n_y}, \mu)
\end{bmatrix}
\]

(4.3)

in which \(\Delta g_{k} = [g_{k} - g_{k-1}, \ldots, g_{k} - g_{k-n_y}]^{T}\).

Therefore, a measurement-based estimation of the operational Jacobian can be implemented as follows:

- **Step 1:** Set a set of initial signal settings \(\{g_{0}, g_{-1}, \ldots, g_{-n_y}\}\); record the corresponding observed flow set \(\{f_{0}^{\text{mea}}, f_{-1}^{\text{mea}}, \ldots, f_{-n_y}^{\text{mea}}\}\); calculate \(\frac{\partial f^{\text{real}}}{\partial g}
\bigg|_{0}\) from (4.3) and input it to the IOCMBC algorithm to compute the initial Jacobian error \(\delta_{0} = \frac{\partial f^{\text{real}}}{\partial g}
\bigg|_{0} - \frac{\partial f^{\text{real}}}{\partial g}
\bigg|_{0}\); then solve \(g_{1}\); set \(k=1\);

- **Step 2:** Apply \(g_{k}\); record signal settings \(\{g_{k}, g_{k-1}, \ldots, g_{k-n_y}\}\) and flows \(\{f_{k}^{\text{mea}}, f_{k-1}^{\text{mea}}, \ldots, f_{k-n_y}^{\text{mea}}\}\); calculate \(\frac{\partial f^{\text{real}}}{\partial g}
\bigg|_{k}\);

- **Step 3:** Input \(\frac{\partial f^{\text{real}}}{\partial g}
\bigg|_{k}\) to IOCMBC algorithm and update \(g_{k+1}\);

- **Step 4:** Stop if the termination condition of IOCMBC is satisfied; otherwise set \(k=k+1\) and go to step 2.

This measurement-based approximation method can estimate the operational Jacobian with sufficient accuracy in the circumstance of measurement noise-free and small number of variables. The impact of its accuracy on the IOCMBC algorithm in a large-scale problem with multiple decision variables is beyond the scope of this paper; while it is an interesting topic for a future study.
5 Numerical example

5.1 Simulation setup

The effectiveness of the proposed IOCMBC method is illustrated in a test network known as the Braess network. As shown in figure 4, this network consists of one OD pair (from node 1 to node 4), 5 links and 3 routes. Link travel cost is calculated using a linearized BPR function. Logit route choice models with the dispersion parameter $\theta$ are utilized for demand modeling. Node 3 represents a signalized intersection operating in a two-phase plan. Green split is the decision variable and signal loss time is not considered in this case thus $g^2 + g^3 = 1$. As one signalized intersection is concerned, a single variable control optimization problem is illustrated in this case study.

Mismatch between model and reality is simulated by using different route choice models. Assume that in reality travelers follow a nested logit structure (Casetta 2009) for making route choice decisions. Regarding link 4 as a highway, it is true that travelers would have more information at node 2 than at node 1 thus separating two choice levels. The probability of choosing route $j$ can be expressed as:

$$
\rho_{real}^j = \frac{\exp(-c_j \theta) \cdot \exp(\zeta Y_j)}{\sum_{i \in l_j} \exp(-c_i \theta) \cdot \sum_h \exp(\zeta Y_h)}
$$

(5.1)

Link travel cost is denoted by $c$. Route choice set is divided into subsets $I_1, ..., I_k, ...$, called nests.

$\zeta = \frac{\theta_0}{\theta}$ is the ratio of dispersion parameters $\theta_0$ and $\theta$ associated with the first and second choice level respectively. $Y_j = \ln \sum_{i \in l_j} \exp(-c_j \theta)$ is the corresponding logsum variable.

In general, this route choice response cannot be precisely described in the demand modeling. We assume that only one choice level is recognized and a multinomial logit model is used with the dispersion parameter $\theta$. Therefore the probability is calculated by the model as:

$$
\rho^j = \frac{1}{1 + \sum_{i \neq j} \exp[\theta(c_j - c_i)]}
$$

(5.2)

As we focus on demand side uncertainty, supply side modeling is assumed to be precise thus the link travel cost is accurately modeled by a linearized BPR function. Introducing a free-flow travel time $c^0$, 

![Figure 4 The Braess test network](https://example.com/figure4.png)
a saturation flow $s$ and a coefficient $\alpha$, the link cost is described by a function of link flow $f$ and signal setting $g$:

$$c = C(f, g) = e^0 + \alpha \frac{f}{gs}$$

Specifically, for non-signalized link, green split $g$ equals 1. The equilibrium flow pattern resulting from the demand-supply interaction depends on the signal control decisions implemented in reality:

$$f^{mea} = f^{real}(g)$$

Total travel time $z$ for this network is also precisely formulated as a function of signal and equilibrium flow:

$$z = z(g, f^{mea}) = \sum_j C(f^{mea}, g_i)f^{mea}$$

A mismatched flow model is simulated following a multinomial logit structure. The dispersion parameter $\theta$ is concerned as an adjustable parameter for the equilibrium flow model:

$$f^{Eq} = f(g, \theta)$$

Control decisions are to be made based on the approximated model, and the objective is to minimize the total travel time on this network. All optimization problems in this numerical example are solved using MATLAB optimization toolbox.

Moreover, OD demand is fixed in this illustrative case and all the links have a same saturation flow. Characteristics of the network are listed in table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OD demand (veh/h)</td>
<td>2000</td>
</tr>
<tr>
<td>Parameter $\alpha$ in cost function</td>
<td>0.15</td>
</tr>
<tr>
<td>Saturation flow (veh/h)</td>
<td>2000</td>
</tr>
<tr>
<td>Free-flow travel time (h)</td>
<td></td>
</tr>
<tr>
<td>link 1</td>
<td>0.1</td>
</tr>
<tr>
<td>link 2</td>
<td>0.2</td>
</tr>
<tr>
<td>link 3</td>
<td>0.05</td>
</tr>
<tr>
<td>link 4</td>
<td>0.4</td>
</tr>
<tr>
<td>link 5</td>
<td>0.15</td>
</tr>
<tr>
<td>Dispersion parameters in reality</td>
<td></td>
</tr>
<tr>
<td>$\theta_0$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\theta$</td>
<td>1</td>
</tr>
</tbody>
</table>

In the equilibrium flow model, dispersion parameter takes a nominal value of $\theta = 10$.

5.2 Results of the numerical experiments

The proposed IOCMBC method is implemented for 20 days in this test network, comparing with a traditional two-step control scheme of iteratively calibrating parameter $\theta$ and optimizing signal settings.
First, by solving the anticipatory control optimization for reality, we have the results of true optimal signal settings, link flows and total travel cost presented in table 2.

Table 2 Optimal solution for reality: signal settings, link flows, total travel time

<table>
<thead>
<tr>
<th>Signal green split</th>
<th>((g^2, g^3) = (0.66, 0.34))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link flows (veh/h)</td>
<td></td>
</tr>
<tr>
<td>Link 1</td>
<td>1033</td>
</tr>
<tr>
<td>Link 2</td>
<td>967</td>
</tr>
<tr>
<td>Link 3</td>
<td>520</td>
</tr>
<tr>
<td>Link 4</td>
<td>513</td>
</tr>
<tr>
<td>Link 5</td>
<td>1487</td>
</tr>
<tr>
<td>Total travel cost (veh.h)</td>
<td>1182.4</td>
</tr>
</tbody>
</table>

As the equilibrium flow model is an imperfect abstraction on the real travelers’ response, implementing control decisions derived from the nominal model-based optimization in real traffic system cannot result in the true optimum. The mismatched results of link flows and total travel cost are shown in table 3.

Table 3 Optimal signal settings derived from model, and implement the model-based signal settings in reality: compare link flows and total travel time

<table>
<thead>
<tr>
<th>Signal green split</th>
<th>Model ((g^2, g^3) = (0.90, 0.10))</th>
<th>Reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link flows (veh/h)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link 1</td>
<td>882</td>
<td>1026</td>
</tr>
<tr>
<td>Link 2</td>
<td>1118</td>
<td>974</td>
</tr>
<tr>
<td>Link 3</td>
<td>285</td>
<td>459</td>
</tr>
<tr>
<td>Link 4</td>
<td>597</td>
<td>567</td>
</tr>
<tr>
<td>Link 5</td>
<td>1403</td>
<td>1433</td>
</tr>
<tr>
<td>Total travel cost (veh.h)</td>
<td>1173.0</td>
<td>1256.3</td>
</tr>
</tbody>
</table>

If a two-step approach is adopted and implemented to deal with the issue of model inaccuracy, it iteratively solves a parameter estimation problem, minimizing the error between measured flows and model prediction flows, and a control optimization problem, minimizing the total travel cost. As shown in figure 5, interaction between two steps generates flip-flopping. The optimal total travel cost cannot be reached. Figure 5(a) shows the flows on link 1 during the 20 days for both model prediction and reality, whereas figure 5(b) depicts the total travel cost for the network. Simply iterating two steps of parameter estimation and control update does not lead to an optimal network performance.
This inconsistency results from the presence of structural model mismatch, which makes an accurate value of the model parameter irrelevant for an accurate system representation. In order to tackle the problem of model mismatch and to integrate the objectives of model correction and performance optimization, our proposed IOCMBC method introduce a reality-tracking metamodel for determining optimal control decisions. Instead of iteratively correcting model parameter, model bias is corrected to track the real network state. IOCMBC is implemented for the test network with large step size (K=1). Figure 6 shows trajectory of the two-step iterates and the IOCMBC iterates starting from the nominal optimal point. Figure 6(a) depicts the total travel cost contours of the true anticipatory control optimization problem. The true optimum is constrained by taking into account route choice behavior. Figure 6(b) compares the iteration trajectories of a conventional two-step approach and the proposed IOCMBC approach. Starting from the nominal point of (green, flow1)=(0.90, 1026), IOCMBC converges to the true optimum in 4 iterations, whereas the flip-flopping between two states occurs under the two-step approach.
As discussed, a key manipulation in IOCMBC is calculation of the operational Jacobian for the real flow response. A perturbation-based finite difference method and a measurement-based finite difference method are adopted and compared in figure 7, with figure 7(a) showing the flows on link 1 and figure 7(b) the total travel costs during the testing days. Both methods converge in this small network. The perturbation-based method, which requires addition input perturbations, and the corresponding equilibrium states can generate more accurate estimations. This is used as a benchmark in the comparison to show the impact of Jacobian estimation on convergence. It is quite natural that the accurate method shows a faster convergence than the measurement-based method. Ongoing work elaborates on the influence of accuracy of the operational Jacobian estimations.

The impact of step size

Now we examine the impact of the step size K performed along the optimization direction to update a new implementation signal setting. Due to the similarity between the nested logit and multinomial logit structure, a convergent control scheme is derived with a full step (i.e. K=1) in calculating the signal control at each iteration. In order to show the role of the step size in regulating convergence, we use another simulation setup and make a further linearization on the equilibrium flow model around each current operation point. In this way, a worse case of model approximation to the reality is considered in which the model varies along the iterations. The reality is still simulated by a nested logit model. A Euclidean distance between the model-based results and the true optimum is used as an indicator for convergence. Denote the Euclidean distance of flow and total travel cost as \( \text{dis}_k^f \) and \( \text{dis}_k^z \) respectively, the distances during iterations are recorded as:

\[
\text{dis}_k^f = \left\| f_k^{\text{mea}} - f_k^{\text{mea}^*} \right\| = \left( \sum_{\text{link}} \left( f_k^{\text{mea}} - f_k^{\text{mea}^*} \right)^2 \right)^{1/2}
\]  

(5.3)
\[ \text{dis}_{k} = \| \mathbf{e}_{k} - \mathbf{z} \| \]  

(5.4)

Figure 8 compares the convergence procedures of IOCMBC with five values for the step size, in terms of the Euclidean distance of flow shown in figure 8(a) and the Euclidean distance of total travel cost shown in figure 8(b). As shown, performing larger step (with \( K \geq 0.5 \)) could lead to non-convergent procedures. It is obvious that selecting a proper step size plays an important role in designing a successfully operating control scheme and thus in regulating the reality-tracking convergence which is our ultimate goal.

Figure 8(a)  

Figure 8(b)  

Figure 8 Convergence of the IOCMBC method with five different step size values: 8(a) Euclidean distance of flows; 8(b) Euclidean distance of total travel cost

6 Discussion

Throughout the discussion on the IOCMBC method, we assume that all link flows are measurable and the measurements are noise-free. However, these assumptions are too rigid regarding applications in large-scale road networks.

Due to a limited set of traffic sensors or missing data in large traffic networks, there is a mismatch between the dimensions of the measured flow \( \mathbf{f}^{\text{meas}}_{k} \) and the modeled flow \( \mathbf{f}_{k}^{\text{Eq}} \) as shown in equation (2.10), when calculating the model bias. As a result, the model bias that is used for the subsequent control optimization cannot be determined from the observed traffic flows. A recently developed methodology for selecting optimal sensor locations has been shown to be effective in tackling the partial observability issue, i.e. when not all flow variables are observed or observable (Viti et al., 2014). The impact of sensor positioning and of partial observability of link flow variables is beyond the scope of this study and is left for future research.

In addition, the presence of measurement noise is detrimental to the accuracy of the operational Jacobian estimates, thus affecting the quality of the IOCMBC solution. The difficulty of estimating Jacobian from past operating points increases with the number of decision variables, as a large
number of measurements are required for implementing the measurement-based finite difference, which typically amplifies the effect of measurement noise. Although the problem of estimating operational Jacobian particularly regarding large-scale applications is crucial to the IOCMBC method, it is beyond the scope of this study. Estimation methods to reduce the impact of measurement error on the Jacobian approximation are to be investigated in a future study.

7 Conclusions

This paper proposes an iterative optimizing control method for designing an adaptive anticipatory control scheme operating in daily traffic, which takes into account route choice behavior and focus on steady-state (equilibrium flow) in traffic signal control design.

It is known that inaccurate model usually fails to generate the true optimal solutions from the model based control optimization problem as the approximated model could not have perfect representation on real system. Furthermore, the presence of structural model mismatch limits the performance of a conventional two-step approach. The proposed method of Iterative Optimizing Control with Model Bias Correction (IOCMBC) allows traffic managers to tackle this limitation and to drive the traffic system towards the true optimal performance despite inevitable model uncertainty.

First, IOCMBC method is formulated based on the idea of handling model bias instead of model parameter. A full first-order correction is performed on model bias at each iteration thus generating a reality-tracking metamodel for iteratively calculating optimal signal settings. We present a theoretical justification on using the IOCMBC method to approximate the true optimal solution. This is achieved by proving the matching of Necessary Optimality Conditions (NOC) and the resulting anticipatory equilibrium flow state for IOCMBC and reality. Moreover, we analyze conditions required for the convergence of IOCMBC. Simulation test in a small network verifies the effectiveness of IOCMBC, as well as the influence of the step size on convergence, which is performed on signal update.

A price to pay for ensuring the true optimality is the need to estimate Jacobian at each iteration. A critical issue of estimating the operational Jacobian for the real traffic flow response is recognized. In order to avoid additional perturbations for deriving the sensitivity information, which is quite impractical, a measurement-based finite difference method is introduced. Influence of accuracy of the operational Jacobian is also examined in our small test network. An ongoing study elaborates estimation methods for handling large-scale problems in the presence of measurement noise.

As discussed, the proposed IOCMBC method actually corrects the reality-tracking metamodel used for determining the optimal control settings. In essence it is a direct control update method, as it improves the network performance iteratively by adjusting the control settings using measurements. The model used to describe reality is not adjusted. Indeed it is not necessary as our goal is to design reality-tracking signal settings and an accurate system description is not the research focus at this point. However, regarding a need for parallel operations in consideration of operational control system design, whereby model calibration is required and the dynamic flow estimation is simultaneously implemented during the process of making optimal control decisions (Han et al., 2010), it is necessary to yield a best system representation as close as possible to the reality at its
current operating point. Under this concern, parameter estimation is performed for a model correction. This framework of iterative model correction and control optimization is a topic for future work.

Finally, in this study we focus on steady-state control at a strategic level based on the argument that the aggregation of individual choice decisions follows an equilibrium pattern. However in some circumstances such as after a bridge collapse (Zhu et al., 2010) day-to-day individual choice behavior would have strong effect on the collective choice behavior. Thus the aggregate day-to-day flow dynamics is activated and transient behavior matters the day-to-day signal control design. The framework of iterative model correction and control optimization allows for the incorporation of transient route choice behavior and the resulting day-to-day flow dynamics. In this case, day-to-day flows are interpreted as parameters to be estimated which are also iteration-varying in reality influenced by signal control. This attractive property and other potentials brought by the iterative framework are appealing and worth further explorations.

Reference


Huang, W., Tampère, C.M.J., Viti, F., 2014. A daily adaptive flow anticipatory control using iterative learning in urban traffic networks. The 5th International Symposium on Dynamic Traffic Assignment (DTA), Salerno, Italy.


Yang, H., Bell, M.G.H., 1998. Models and algorithms for road network design: a review and some new developments. TRANSPORT REVIEWS, VOL. 18, No. 3, 257-278.

