https://doi.org/10.1016/j.trc.2020.102618

Data Driven Model Free Adaptive Iterative Learning Perimeter Control for Large-scale Urban Road Networks

Ye Ren[a], Zhongsheng Hou[c,a,*], Isik Ilber Sirmatel[b], Nikolas Geroliminis[b]

[a] *Advanced Control Systems Laboratory, Beijing Jiaotong University,100044 Beijing, China*
[b] *Urban Transport Systems Laboratory, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*
[c] *School of Automation, Qingdao University, 266071 Qingdao, China*

Abstract

Most perimeter control methods in literature are the model-based schemes designing the controller based on the available accurate macroscopic fundamental diagram (MFD) function with well known techniques of modern control methods. However, accurate modeling of the traffic flow system is hard and time-consuming. On the other hand, macroscopic traffic flow patterns show heavily similarity between days, and data from past days might enable improving the performance of the perimeter controller. Motivated by this observation, a model free adaptive iterative learning perimeter control (MFAILPC) scheme is proposed in this paper. The three features of this method are: 1) No dynamical model is required in the controller design by virtue of dynamic linearization data modeling technique, i.e., it is a data-driven method, 2) the perimeter controller performance will improve

*Corresponding author
Email addresses: 14111048@bjtu.edu.cn (Ye Ren), zshou@qdu.edu.cn (Zhongsheng Hou), isik.sirmatel@epfl.ch (Isik Ilber Sirmatel), nikolas.geroliminis@epfl.ch (Nikolas Geroliminis)

iteratively with the help of the repetitive operation pattern of the traffic system, 3) the learning gain is tuned adaptively along the iterative axis. The effectiveness of the proposed scheme is tested comparing with various control methods for a multi-region traffic network considering modeling errors, measurement noise, demand variations, and time-changing MFDs. Simulation results show that the proposed MFAILPC presents a great potential and is more resilient against errors than the standard perimeter control methods such as model predictive control, proportional-integral control, etc.
Keywords: Macroscopic fundamental diagram (MFD), model free adaptive control (MFAC), perimeter control, large-scale urban road networks

1. Introduction

Urban traffic signal control is widely used to alleviate congestion all over the world. In the past decades, a number of methods for urban traffic control systems have been developed, such as SCOOT (Robertson and Bretherton, 1991), SCATS (Lowrie, 1982), TUC (Diakaki et al., 2002), and max pressure (Varaiya, 2013). However, accuracy and computational burden challenge the possible successful application of these optimization and control methods due to the requirement of the detailed models of the controlled urban networks. Moreover, these controllers make decisions at a local level without considering network effects, which may lead to suboptimal solutions due to congestion propagation. Therefore, developing an aggregated model for large-scale road networks is essential for both efficiency and reliability reasons. The macroscopic fundamental diagram (MFD) of urban traffic networks attracts increasing interest in literature owing to its potential in enabling the devel-

opment of low complexity network-level traffic models, and ultimately the design of real-time management systems for large-scale urban networks.

The idea of MFD was first proposed in Godfrey (1969), but its existence with dynamic features has not been verified until 2008 by the empirical data of Yokohama (Geroliminis and Daganzo, 2008). Thereafter there has been an increased interest in MFD as numerous contributions have been made including theoretical analyses of the MFD, region partitioning via clustering, network-wide traffic modeling and control system design via actuation over perimeter control. Nevertheless, the heterogeneous distribution of congestion and hysteresis phenomena (Geroliminis and Sun, 2011; Gayah and Daganzo, 2011; Mahmassani et al., 2013) may affect the presence of a well-defined MFD. To overcome these drawbacks, one alternative way is to partition the large heterogeneous network into several homogeneous sub-regions. Various clustering algorithms for static and dynamic configuration have been developed to network partitioning, such as (Ji and Geroliminis, 2012; Saeedmanesh and Geroliminis, 2016, 2017; Lopez et al., 2017; Casadei et al., 2018). Clustering provides the possibility to build multi-region MFD models that can be integrated in perimeter control.

Based on MFD-based dynamical models, traffic control schemes for the perimeter control (i.e., regulating a set of traffic lights on the boundary between two neighbouring regions) are proposed to improve urban traffic conditions. The application of perimeter control to single region networks can be found in Daganzo (2007); Keyvan-Ekbatani et al. (2012, 2015a); Haddad and Shraiber (2014); Haddad (2017a). In Keyvan-Ekbatani et al. (2012), a proportional-integral (PI) feedback controller is involved. These results are

extended to the case with time-delays in Keyvan-Ekbatani et al. (2015a). Considering traffic system uncertainty, a robust PI perimeter controller is designed by transforming the plant into a linear parameter-varying model in Haddad and Shraiber (2014). In Haddad (2017a), the optimal control laws are proposed for both coupled and decoupled perimeter control inputs. Perimeter control is also applied to two-region urban networks with MFD-based dynamics. The two-region perimeter control problem is first formulated in Haddad and Geroliminis (2012) and stability conditions for stable equilibrium are derived. Boundary conditions and controllability issues are investigated in Zhong et al. (2018). In Geroliminis et al. (2013); Haddad (2017b), model predictive control (MPC) is used to design the perimeter controller for two-region systems.

There are also recent works focusing on multi-region MFD networks. In Aboudolas and Geroliminis (2013), a multivariable linear quadratic regulator (LQR) is designed by linearizing the model at an equilibrium point. A practical multiple concentric PI controller is proposed in Keyvan-Ekbatani et al. (2015b). In order to adapt to different traffic conditions, the controller in Aboudolas and Geroliminis (2013) is enhanced by online adaptive optimization in Kouvelas et al. (2017), which is the first study in the literature where gains are updated based on historical data and performance of the controller from day to day. MPC is also applied to multi-region networks in Ramezani et al. (2015); Fu et al. (2017); Zhou et al. (2017); Sirmatel and Geroliminis (2018). In Haddad and Mirkin (2016, 2017), the centralized and distributed adaptive perimeter control approaches are proposed, respectively. In the recent work Lei et al. (2019), a decentralized estimation

and decentralized MFAC method is applied to perimeter control to solve the strong-coupled interconnections between sub-regions. The other kinds of MFD-based modeling and perimeter control methods could also be found in: integration of agent-based modeling with MFD (Kim et al., 2018a), modeling of macroscopic flows considering link level capacity (Kim et al., 2018b), hybrid PI control (Ding et al., 2018), control via vehicle routing (Menelaou et al., 2017), optimal control (Aalipour et al., 2018), congestion pricing in a connected vehicle environment (Yang et al., 2019), uncertainty modeling (Gao and Gayah, 2018) and robust control (Zhong et al., 2017; Ampountolas et al., 2017; Mohajerpoor et al., 2019).

However, almost all of the aforementioned methods are model-based. Due to the inevitable unmodelled dynamics and unknown demand uncertainty, the mismatch between the model predictions and the plant is a potential source of problems for model-based perimeter control. For this reason, it is worth investigating perimeter control methods independent of the dynamical model. On the other hand, traffic flow patterns usually show similarity from a macroscopic point of view (Hou and Xu, 2003; Hou et al., 2008; Hou and Li, 2016), even if individual roads might have strong daily variations. In such circumstances, the performance of the controller would improve greatly by making full use of the repeatability and similarity of a traffic system.

Iterative learning control is an effective method to deal with repeated control processes. It was originally proposed in Arimoto et al. (1984), and has been extensively developed in the past 30 years with various learning controllers including proportional derivative (PD) type ILC (Saab, 2003), norm optimal ILC (Amann et al., 1996) and Lyapunov-based adaptive ILC

(Tayebi, 2004). In Hou and Xu (2003), ILC is first applied to the traffic system to solve the density control problem of freeways. For more detailed reviews of ILC, readers are pleased to read the survey papers Bristow et al. (2006); Xu (2011). Although ILC is widely studied, some fundamental limitations still exist. Controller design of the norm optimal or Lyapunov-based adaptive ILC requires model dynamics, and the selection of the learning gain of PD-type ILC is nontrivial. Recently, data-driven model-free adaptive ILC (MFAILC) (Chi and Hou, 2007; Chi et al., 2015; Bu et al., 2019) has been proposed on a new type of dynamic linearization (DL) data modeling technique (Hou and Jin, 2013). Distinguished from the existing other linearization methods, such as Taylor's, input-output linearization method, etc.(Hou et al., 2017), the (DL) data modeling technique does not linearize the system at a fixed point but builds an equivalent linearization data model at each operation point. Using only input and output (I/O) data, all the parameters of this dynamic linearization data model can be estimated and then the controller is designed. Considering different memory lengths of the considered system, the DL data model can be classified into three forms: the compact-form dynamic linearization (CFDL) data model, the partial-form dynamic linearization (PFDL) data model, and the full-form dynamic linearization (FFDL) data model (Hou and Jin, 2013). By virtue of the DL data modeling technique, two critical assumptions, identical initial condition and identical trajectory, to traditional ILC, can be relaxed in the MFAILC method. It is also worthwhile pointing out that MFAILC is a branch of model free adaptive control (MFAC), which has been developed into a systematic framework in the past 20 years (see Liu and Yang (2019); Wu et al. (2018);

Hou and Jin (2013); Hou et al. (2017); Hou and Xiong (2019) for some recent progress of MFAC). Until now, MFAC has been applied successfully to over 150 different fields (see Hou and Wang (2013); Hou et al. (2017) for the survey of MFAC approaches). Since the urban traffic network has following outstanding features, that is, repetitive operation pattern is obvious, the traffic system modeling is difficult, and the traffic system data is easy to be obtained, these features motivate us using the MFAILC method in this paper to deal with the perimeter control problem.

The main contributions are as follows.

1) The time domain DL description used in Lei et al. (2019) is extended the multi-input and multi-output (MIMO) iterative form, which presents the relationship of multi-region urban traffic network dynamics between successive iterations.

2) Based on the novel iterative DL description, a data-driven model free adaptive iterative learning perimeter control (MFAILPC) strategy is employed in the multi-region urban traffic network in the first time. In particular, the proposed method is compared with other several typical perimeter algorithms in a comprehensive way.

Three attractive properties of this method are as follows.

1) The data-driven feature: the controller design and implementation are independent from the traffic plant model, i.e., the system modeling and regional demand profiles are not needed in the perimeter control. Instead, it only requires I/O data of the traffic network, which means complication due to plant-model mismatch is avoided. Note that the critical accumulation used in the algorithm is an input to the method, which is separated from the

controller design and not related to the data-driven feature.

2) The iterative learning feature: the controller obtains "experience" from the historical data stored in the data base. In other words, the perimeter control performance will improve over iterations.

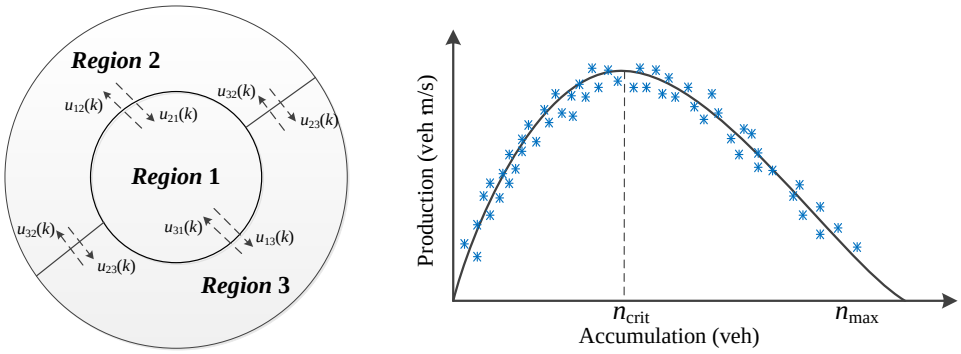
3) The adaptive feature: by estimating the pseudo Jacobian matrix, the learning gain is updated iteratively, which implies the learning gain selection problem in the tradition ILC method disappears.

The rest of the paper is organized as follows. In Section II, the urban traffic system dynamics is described using MFD. In Section III, the MFAILC based perimeter control strategy is introduced in detail. In Section IV, simulation results compared with other typical perimeter control strategies are given. The main conclusions and future works are summarized in Section V.

2. Dynamics for Multi-Region Urban Traffic Network System

The aggregated models (Ramezani et al., 2015; Sirmatel and Geroliminis, 2018) for the urban network considering inflow demands and transfer flows are presented in the following part. Note that the traffic model here is only used to simulate traffic reality and generate traffic data, and not for the perimeter controller design.

Consider a network $\mathscr{R}$ with heterogeneous distribution of congestion, consisting of R homogeneous regions, i.e., $\mathscr{R} = \{1, 2, ..., R\}$, each with a well-defined MFD (see Fig 1 for the representative three-region network and MFD). From the mass conservation equations for urban traffic, one gets the



(a) (b)

Figure 1: (a) Schematic diagram of the urban network with three regions; (b) Schematic diagram of a well-defined MFD.

following dynamic evolution of the i-th ($i \in \mathscr{R}$) region.

$$n_i(t) = \sum_{j \in \mathscr{R}} n_{ij}(t) \tag{1}$$

$$\dot{n}_{ii}(t) = d_{ii}(t) + \sum_{h \in \mathscr{N}_i} u_{hi}(t) M_{hii}(t) - M_{ii}(t) \tag{2}$$

$$\dot{n}_{ij}(t) = d_{ij}(t) + \sum_{h \in \mathscr{N}_i; j \neq h} u_{hi}(t) M_{hij}(t) - \sum_{h \in \mathscr{N}_i} u_{ih}(t) M_{ihj}(t) \ (j \in \mathscr{R}) \tag{3}$$

where $n_i(t)$ (veh) is the accumulation of region i at time step t, $n_{ij}(t)$ (veh) is number of vehicles in region i with destination region j, $d_{ij}(t)$ (veh/s) is the inflow demand from region i to destination region j, $u_{ih}(t) \in [u_{min}, u_{max}]$ (with $0 \leq u_{min} < u_{max} \leq 1$) is the perimeter control input (i.e., the percentage of flow allowed to transfer from region i to region h), $M_{ihj}(t)$ (veh/s) is the transfer flow from region i to destination j through the next immediate region h, $M_{ii}(t)$ (veh/s) is internal trip completion flow (i.e., exit flow). The transfer flow terms are defined as:

$$M_{ihj}(t) = \theta_{ihj}(t) \cdot \frac{n_{ij}(t)}{n_i(t)} \cdot \frac{P_i(n_i(t))}{l_{ij}} \quad (h \in \mathscr{N}_i, i \neq j) \tag{4}$$

$$M_{ii}(t) = \frac{n_{ii}(t)}{n_i(t)} \cdot \frac{P_i(n_i(t))}{l_{ii}} \tag{5}$$

where $P_i(n_i(t))$ (veh·m/s) is the production MFD of region i, l_{ij} (m) is the average trip length inside region i for vehicles with destination j, and $\theta_{ihj}(t) \in [0,1]$ (with $\sum_{h \in \mathscr{N}_i} \theta_{ihj}(t) = 1$) is the route choice term expressing the ratio of flow that is transferring to the next immediate region h over the total flow exiting region i with destination j.

For a multi-region MFD newtork, the route choice terms $\theta_{ihj}(t)$ are calculated by a logit model (see Ben-Akiva and Bierlaire (1999)) using the instantaneous travel times of K-shortest paths found by Dijkstra's algorithm (see Sirmatel and Geroliminis (2018)). The functional form of the MFD can be obtained by processing traffic data in space average and fitting a given curve (Geroliminis and Daganzo, 2008). In this paper, $P_i(n_i(t))$ is expressed by a third-order polynomial function:

$$P_i(n_i(t)) = a_i n_i^3(t) + b_i n_i^2(t) + c_i n_i(t) \tag{6}$$

where a_i, b_i, c_i are parameters extracted from historical traffic data.

The transfer flow between two adjacent regions is restrained by the boundary capacity (Ramezani et al., 2015; Sirmatel and Geroliminis, 2018), which takes the following form:

$$\hat{M}_{ihj}(t) = min\left(M_{ihj}(t), C_{ih}(n_h(t)) \frac{M_{ihj}(t)}{\sum_{o \in \mathscr{R}} M_{iho}(t)} \right) \tag{7}$$

where $\hat{M}_{ihj}(t)$ is the capacity-restrained transfer flow, and $C_{ih}(n_h(t))$ is the boundary capacity between region i and h, which can be modeled as a func-

tion of the accumulation $n_h(t)$:

$$C_{ih}(n_h(t)) = \begin{cases} C_{ih}^{max} & \text{if } 0 \leq n_h(t) \leq \alpha \cdot n_h^{jam} \\ \frac{C_{ih}^{max}}{(1-\alpha)} \cdot (1 - \frac{n_h(t)}{n_h^{jam}}) & \text{if } \alpha \cdot n_h^{jam} \leq n_h(t) \leq n_h^{jam}, \end{cases} \tag{8}$$

where C_{ih}^{max} is the maximum value of boundary capacity between region i and region h, n_h^{jam} is the jam accumulation of region h, and $\alpha \in (0, 1)$ is an estimated parameter.

3. Methodology Framework

3.1. Iterative Dynamic Linearization for Traffic Dynamics

Define following collective vectors:

$$\mathbf{n}(k) = [n_1(k), n_2(k), ..., n_R(k)]^T$$

$$\mathbf{u}(k) = [u_{12}(k), u_{21}(k), ..., u_{ij}(k), u_{ji}(k), ..., n_{(R-1)R}(k), n_{R(R-1)}(k)]^T$$

$$\mathbf{d}(k) = [d_{12}(k), d_{21}(k), ..., d_{ij}(k), d_{ji}(k), ..., d_{(R-1)R}(k), d_{R(R-1)}(k)]^T$$

where $i, j \in \mathscr{R}, i \neq j$, and $\mathbf{n}(k) \in \mathbb{R}^R, \mathbf{u}(k) \in \mathbb{R}^{R\cdot(R-1)}, \mathbf{d}(k) \in \mathbb{R}^{R\cdot(R-1)}$ stand for the vectors contain all the accumulation, perimeter control input, and demand data at sampling instant k, respectively.

For controller design purposes, only the information at the current control time instant is required. To distinguish data from different iterations, an extra indicator l is added to the formulation to present the iteration number, which dedicates to present the different days, or different weeks, or even months. Thus, the plant (1)-(8) is rewritten as the general form of the discrete-time nonlinear system.

$$\mathbf{n}(k+1, l) = \mathbf{f}(\mathbf{n}(k, l), \mathbf{u}(k, l), \mathbf{d}(k, l)) \tag{9}$$

where $\mathbf{f}(\cdot)$ is the corresponding vector-valued nonlinear function, and $\mathbf{n}(k,l) \in \mathbb{R}^R$, $\mathbf{u}(k,l) \in \mathbb{R}^{R\cdot(R-1)}$, $\mathbf{d}(k,l) \in \mathbb{R}^{R\cdot(R-1)}$ are the data vectors at time instant k for iteration l, respectively.

Note that the detailed form of $\mathbf{f}(\cdot)$ is not needed, because it will soon be transformed into an equivalent linearization data model description. Before the linearization, some assumptions are given. Let us consider a finite sampling time interval $\{0, 1, ..., T\}$ between different iterations (e.g., 7 to 9 A.M. in different days). The system (9) is assumed to satisfy following assumptions.

Assumption 1: The partial derivatives of $\mathbf{f}(\cdot)$ with respect to each component of the control vector $\mathbf{u}(k,l)$ are continuous.

Assumption 2: For all $k \in \{0, 1, ..., T\}, l = 1, 2, ...,$ and $\|\Delta\mathbf{u}(k,l)\| \neq 0$, the generalized Lipschitz condition is satisfied for the system (9) when

$$\|\Delta\mathbf{n}(k+1,l)\| \leq b\|\Delta\mathbf{u}(k,l)\|$$

where $\Delta\mathbf{n}(k+1,l) = \mathbf{n}(k+1,l) - \mathbf{n}(k+1,l-1)$, $\Delta\mathbf{u}(k,l) = \mathbf{u}(k,l) - \mathbf{u}(k,l-1)$, $\|\cdot\|$ denotes the Euclidean norm for a given vector and b is a positive constant.

Remark 1: Considering practical situations, the validity of the above assumptions has been widely discussed in Chi and Hou (2007); Chi et al. (2015); Bu et al. (2019). Assumption 1 is a general condition for controller design and easy to verify from (2) and (3). Assumption 2 means that changes in accumulations caused by changes in perimeter control inputs are bounded, which is valid since in reality the traffic network accumulation change cannot be infinite due to finite changes in perimeter control inputs.

In order to design the perimeter controller for a traffic network with multi-regions, the iterative CFDL data model in Chi and Hou (2007) is extended

to a general MIMO system. The main result is summarized as the following theorem.

Theorem 1: Consider the urban traffic network with dynamics (9) satisfying Assumptions 1 and 2. If $||\Delta \mathbf{u}(k,l)|| \neq 0$, then system (9) can be equivalently transformed into an iterative CFDL data model for all $k \in \{0,1,...,T\}, l = 1,2,...,$

$$\Delta \mathbf{n}(k+1,l) = \mathbf{\Phi}(k,l)\Delta \mathbf{u}(k,l) \tag{10}$$

where $\mathbf{\Phi}(k,l) = [\phi_{ij}(k,l)] \in \mathbb{R}^{R\times(R\cdot(R-1))}$ is an iteration-varying and time-varying matrix named pseudo Jacobian matrix (PJM) which is bounded for any time instant k and iteration l.

Proof: See Appendix A.

Remark 2: In theorem 1, the traffic dynamics is equivalently transformed into a dynamic incremental form data model (10) with PJM along the iteration axis. This description avoids some imprecise problems in other existing linearization methods, such as the loss of high order terms in Taylor's linearization (Chen and Narendra, 2004), and the requirement of the model information in piecewise linearization like the switching time (Xi et al., 1996). Theorem 1 is derived under the condition $\|\Delta \mathbf{u}(k,l)\| \neq 0$. If the case $\|\Delta \mathbf{u}(k,l)\| = 0$ happens at some sampling time or iteration number, a new CFDL data model can also be established after shifting $\gamma_i \in \mathbb{Z}^+$ iterations until $\mathbf{u}(k,l) \neq \mathbf{u}(k,l+\gamma_i)$ holds. By virtue of the sensor technology, the traffic network have abundant I/O data to estimate PJM. In the next section, this data model will be used to design the perimeter control scheme.

3.2. Model Free Adaptive Iterative Learning Perimeter Controller Design

The perimeter controller design consists of two steps. First, define the cost functions for the PJM estimation and perimeter control. A cost function minimizing the error between the real system output and CFDL model output is proposed as follows.

$$\begin{aligned} J(\mathbf{\Phi}(k,l)) = &\|\Delta\mathbf{n}(k+1,l-1) - \mathbf{\Phi}(k,l)\Delta\mathbf{u}(k,l-1)\|^2 \\ &+ \ \mu\xi^2\|\mathbf{\Phi}(k,l) - \hat{\mathbf{\Phi}}(k,l-1)\|^2 \end{aligned} \tag{11}$$

where $\hat{\mathbf{\Phi}}(k,l-1)$ is the estimated value of PJM at time instant k for iteration $l-1$, ξ is a normalizing factor used to balance the magnitude of the two terms and μ is a weighting factor to constrain the change of estimated value between successive iterations.

The objective function of the perimeter control inputs is proposed as follows.

$$J(\mathbf{u}(k,l)) = \|\mathbf{n}_{crit} - \mathbf{n}(k+1,l)\|^2 + \lambda\zeta^2\|\Delta\,\mathbf{u}(k,l)\|^2 \tag{12}$$

where ζ is a normalizing factor to balance the magnitude of the two terms, λ is a weighting factor to constrain the change of control input between successive iterations, $\mathbf{n}_{crit} = [n^1_{crit}, n^2_{crit}, ..., n^R_{crit}]$ is the vector containing all the critical accumulation and n^i_{crit} denotes the critical accumulation for the ith region. Since the outflow of the network is maximized at the critical point, the objective function (12) represents the goal of maximizing the outflow of all regions in the network.

Second, the control scheme is derived by substituting (10) into (12) and

minimizing the above two cost functions:

$$\hat{\mathbf{\Phi}}(k,l) = \hat{\mathbf{\Phi}}(k,l-1) + \eta \left(\Delta \mathbf{n}(k+1,l-1) - \hat{\mathbf{\Phi}}(k,l-1) \Delta \mathbf{u}(k,l-1) \right) \times \frac{\Delta \mathbf{u}^T(k,l-1)}{\mu \xi^2 + \|\Delta \mathbf{u}(k,l-1)\|^2} \tag{13}$$

$$\mathbf{u}(k,l) = \mathbf{u}(k,l-1) + \rho \frac{\hat{\mathbf{\Phi}}^T(k,l) \left(\mathbf{n}_{crit} - \mathbf{n}(k+1,l-1) \right)}{\lambda \zeta^2 + \|\hat{\mathbf{\Phi}}(k,l)\|^2} \tag{14}$$

where $\eta \in (0,2]$ and $\rho \in (0,1]$ are step factors added to make (13) more general. Note that the matrix inversion has been replaced by a simple projection mechanism in (13)-(14) for implementation simplicity.

Considering the physical constraints $0 \leq u_{min} \leq u_{ij}(k,l) \leq u_{max} \leq 1$ on the control inputs, each component $u_{ij}(k,l)$ in $\mathbf{u}(k,l)$ is then modified to satisfy the constraints:

$$\bar{u}_{ij}(k,l) = \begin{cases} u_{min}, & \text{if } u_{ij}(k,l) < u_{min} \\ u_{max}, & \text{if } u_{ij}(k,l) > u_{max} \text{ or} \\ & n_i(k,l) < n^i_{crit} \; and \; n_j(k,l) < n^j_{crit} \\ u_{ij}(k,l), & \text{otherwise} \end{cases} \tag{15}$$

where $\bar{u}_{ij}(k,l)$ is the actual control input, u_{min} and u_{max} are the minimum and maximum values of the perimeter control input, respectively. In addition, there is no need to restrain the perimeter control input between two uncongested neighboring region i and region j (i.e., $n_i(k,l) < n^i_{crit}$ *and* $n_j(k,l) < n^j_{crit}$). Here, the eq. (15) serves as the constraint after the calculation of (14) for the simplicity of practical implementation. And all constraints can also be put into one optimization which is discussed in Remark 4.

In summary, the MFAILPC perimeter controller implementation procedure can be schematically shown in Figure 2 and formalized via the algorithm

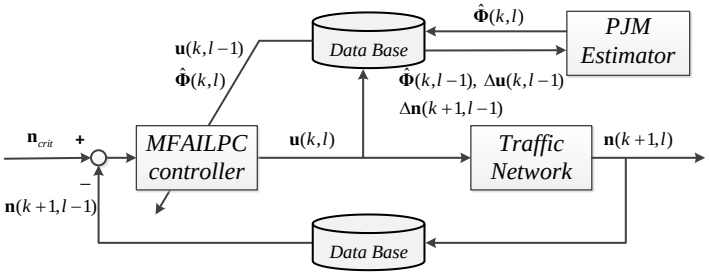


Figure 2: Schematic diagram of MFAILPC.

1, where T_l is the final time step and l_{final} is the preset maximum iteration number.

Algorithm 1 Model free adaptive iterative learning perimeter control

1) Set l=1 and initialize the database.
2) Set k=1 and initialize controller parameters.
3) Estimate JPM $\hat{\mathbf{\Phi}}(k,l)$ using (13).
4) Calculate control input $\mathbf{u}(k,l)$ using (14) and (15).
5) Store new data generated in time step k and iteration l in the database.
6) If $k < T_l$, set $k = k+1$ and go to step 3. Otherwise, go to step 7.
7) If $l < l_{final}$, set $l = l+1$, and go to step 2. Otherwise, end the procedure.

Remark 3: From (13) and (14), one can see that all the information used in the algorithm can be measured from the system. For instance, if one wants to update the estimated PJM $\hat{\mathbf{\Phi}}(k,l)$, all the information needed is historical data from previous iterations, i.e., $\mathbf{u}(k,l)$, $\mathbf{u}(k,l-1)$, $\mathbf{n}(k+1,l-1)$, $\mathbf{n}(k+1,l-2)$ and $\hat{\mathbf{\Phi}}(k,l-1)$. Moreover, the critical accumulation $\mathbf{n}_{crit}$ can also be derived from the I/O data since the MFD is obtained by fitting the historical traffic data spatially aggregated (Geroliminis and Daganzo,

2008). Thus, it is a pure data-driven method as no mathematical model is involved in the algorithm. In addition, it can be observed from (14) that the learning gain is $\hat{\mathbf{\Phi}}^T(k,l)/\left(\lambda\zeta^2+\|\hat{\mathbf{\Phi}}(k,l)\|^2\right)$ which varies with recently estimated PJM. Therefore, MFAILPC also possesses the adaptive feature to tune the learning gain along the iterative axis.

Remark 4: In (14), the perimeter control input is calculated by an iterative formula. Here, the control constraint can also be taken into consideration by solving the following optimization problem.

$$\begin{aligned} &\underset{\mathbf{u}(k,l)}{\text{minimize}} \ ||\mathbf{n}_{crit}-\mathbf{n}(k+1,l-1)-\hat{\mathbf{\Phi}}(k,l)\Delta\,\mathbf{u}(k,l)||^2+\lambda\zeta^2||\Delta\,\mathbf{u}(k,l)||^2 \\ &\text{subject to} \qquad\qquad \mathbf{g}\left(\mathbf{u}(k,l),\mathbf{u}_{min},\mathbf{u}_{max},\Delta\mathbf{u}_{max}\right)\leq 0 \end{aligned} \tag{16}$$

where $\mathbf{g}(\cdot)$ is the corresponding vector-valued function derived from the constraints, $\mathbf{u}_{min}$, $\mathbf{u}_{max}$ and $\Delta\,\mathbf{u}_{max}$ are vectors contains all lower bound u_{min}, upper bound u_{max}, and the bound of the input change rate, respectively. It is easy to verify that problem (16) is a convex quadratic programming (QP) problem, which can be easily and efficiently solved by the convex QP solver.

Remark 5: The parameter tuning rules of MFAILPC are as follows: the normalizing factors ξ and ζ are selected according to the magnitude of the system input and output (e.g.,the normalizing factors should be around 5000 if $n_i(k)\in[0,5000]$ and $u_{ij}(k)\in[0,1]$). The entries of initial PJM $\hat{\mathbf{\Phi}}(k,1)$ are chosen as small constants and sign of them can be decided by the corresponding input (e.g., $\phi_{21}(k,1)$ should be positive if $n_1(k)$ tends to increase when $\Delta u_{21}(k)$ is a positive value). η and μ are chosen to satisfy $\eta\in(0,\ 2]$ and $\mu>0$ in order to guarantee that the PJM estimation in (13) is bounded (Hou and Jin, 2013; Chi et al., 2015). From the existing theoretical analysis,

it shows there exists a constant λ_{min} and the stability of the MFAILC control scheme can be guaranteed if one chooses $\lambda > \lambda_{min}$ (Chi and Hou, 2007; Hou and Jin, 2013; Chi et al., 2015; Bu et al., 2019).

3.3. Comparative Perimeter Control Strategies

Other four typical control schemes are selected as comparison. A brief introduction of each method is presented below and a summary is given in Table 2

1) No Control (NC)

As a benchmark for comparison, the transfer flow in the NC strategy will be kept to the original value and all the perimeter control inputs are fixed to their maximum value of 1, i.e., $u_{ij} = 1,\ (i, j \in \mathscr{R})$.

2) P-type Iterative Learning Control (P-type ILC)

To compare with the adaptive feature of MFAILPC, a P-type ILC stategy (Saab, 2003) is designed with a fixed learning gain matrix as follows:

$$\mathbf{u}(k, l) = K_{ILC} * (\mathbf{n}_{crit} - \mathbf{n}(k + 1, l - 1)) \tag{17}$$

where $K_{ILC} \in \mathbb{R}^{6\times3}$ is the learning gain matrix. In the simulation, this gain is chosen by a trial and error method according to the performance.

3) Feedback PI Control

As the most commonly used control method, the PI controller is from (Keyvan-Ekbatani et al., 2015b). However, the traditional PI controller only gets the feedback from the current time step and can never improve its performance no matter how many times the process runs. In this test, the PI controller is mainly used to present the difference between controllers with

and without learning ability. The controller is designed as:

$$\mathbf{u}(k) = K_P * (\mathbf{n}_{crit} - \mathbf{n}(k)) + K_I * \sum_{\nu=0}^{k} (\mathbf{n}_{crit} - \mathbf{n}(\nu)) \tag{18}$$

where K_P and K_I are feedback gain matrices. In the simulation part, K_P and K_I are tuned following a trial and error method. And the final values are chosen from the best result after 50 times of trials.

4) Model Predictive Control (MPC)

Model predictive control (MPC) is an advanced model-based control technique using real-time rolling optimization. However, it also has limitations since its performance depends on the accuracy of the dynamical model and demand information. In this paper, the comparison between MFAILPC and MPC is to verify if the data-driven controller can compete with a model-based one with full knowledge of the plant. The MPC strategy is conducted according to the steps in (Mayne et al., 2000; Sirmatel and Geroliminis, 2018)

5) Decentralized Estimation and Decentralized MFAC (DED-MFAC)

For comparison purpose, the DED-MFAC scheme in Lei et al. (2019) is also applied to the simulation. The main difference between DED-MFAC and MFAILPC are two folded. First, the proposed MFAILPC has the iterative learning ability. In a repetitive traffic environment, MFAILPC will get higher performance over iterations. Conversely, DED-MFAC does not have this feature. Second, MFAILPC is a centralized method based on the MIMO DL description, which gives the algorithm more comprehensive consideration for all regions and will have better performance. DED-MFAC is a decentralized method, which has flexible structures but lose some performance.

4. Case Studies

To test the performance of the proposed method, six cases with different types of uncertainties are simulated on a three-region network (as the one in Fig. 1(a)). The cases are built from the simple to the complex: Case 1 is a basic one without errors; In Case 2-4, each case considers one typical uncertain factor with different levels; Case 5-6 give results under more realistic situations. More details are summarized in Table 1. It is worth noting that, for the proposed MFAILPC method, the plant is used only as I/O data generator and not involved in the controller design.

Table 1: Case Settings

	Error types	Purposes	Network Parameters
Case I	no error	a benchmark and illustration of the learning process	$a_i = 1.4877 \cdot 10^{-7}$ $b_i = -2.9815 \cdot 10^{-3}$ $c_i = 15.091$, $n_{ij}(0) = 100$ (veh) $n^i_{jam} = 10000$ (veh) $n^i_{crit} = 3400$ (veh) $(i, j \in \mathscr{R})$ $l_{11} = l_{22} = l_{33} = 3600$ (m) $l_{12} = l_{13} = 3600$ (m) $l_{21} = l_{31} = 3000$ (m) $l_{23} = l_{32} = 4200$ (m) $\alpha = 0.64$, $C^{max}_{ij} = 3.2$ (veh/s) $u_{min} = 0.1$, $u_{max} = 1$
Case II	MFD scaling error	test the performance under MFD modeling uncertainties	
Case III	measurement noise	test the performance under noise generated from sensors	
Case IV	iteration-varying demands	test the performance under non-strict repetitive environments	
Case V	mixed errors	a special case containing the errors in Case II-IV	
Case VI	time-changing MFDs	test the performance under time-changing traffic networks	

4.1. Network Setup and Parameter Settings

All the cases are simulated in a three-region network as shown in Fig. 1(a), and each region is assumed to have the same MFD observed in a part of downtown Yokohama (Geroliminis and Daganzo (2008)). The sampling

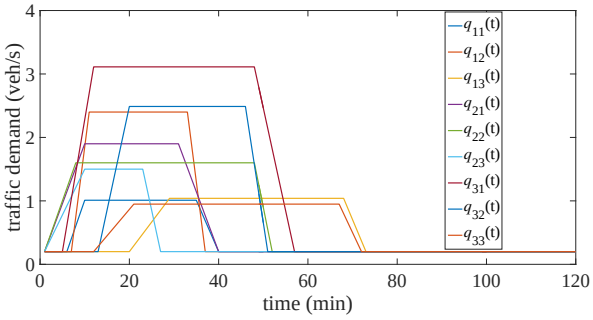


Figure 3: Traffic demand profiles.

interval is chosen as $T_s = 60$ s which is a realistic cycle time for intersections. The total length of the simulation is $T_l = 120$ in number of time steps, representing a duration of 2 hours. The demand pattern is depicted in Fig. 3, which shows that the demand to region 1 (i.e., city center) is higher than other regions (see q_{31}(t) for example). Therefore, this demand profile can represent the traffic situation during morning peak. Other parameters of the network are listed in Table 1.

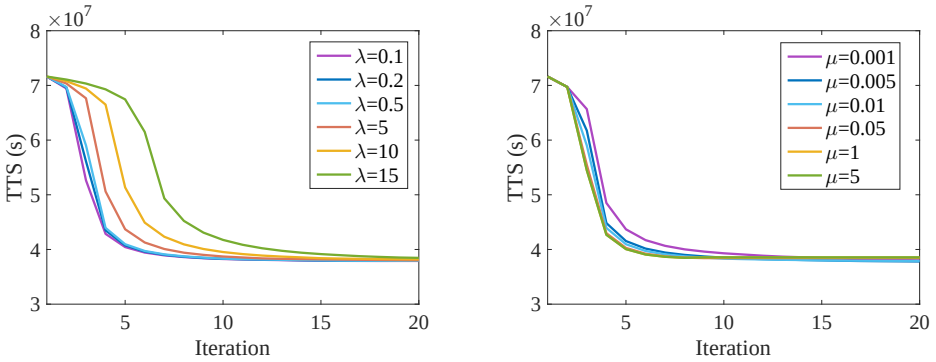
Four performance criteria including total time spent (TTS), time spent per vehicle (TSPV), total network throughput (TNT), and network throughput percentage (NTP) are defined as follows:

$$TTS = T_s \cdot \sum_{k=1}^{T_l} \sum_{i \in \mathscr{R}} n_i(k), \qquad TSPV = \frac{\sum_{k=1}^{T_l} \sum_{i \in \mathscr{R}} n_i(k)}{\sum_{i,j \in \mathscr{R}} d_{ij}(k)}$$

$$TNT = T_s \cdot \sum_{k=1}^{T_l} \sum_{i \in \mathscr{R}} M_{ii}(k), \quad NTP = \frac{\sum_{k=1}^{T_l} \sum_{i \in \mathscr{R}} M_{ii}(k)}{\sum_{i,j \in \mathscr{R}} d_{ij}(k)} \times 100\%$$

Table 2: Control Strategy Summary

Method	Controller Description	Parameters	Purpose
No Control	$u_{ij} = 1,\ (i,j \in \mathscr{R})$	-	benchmark
MFAILPC	eq. (13), (14), and (15)	$\lambda = 0.5,\ \mu = 0.01,\ \xi = \zeta = 5000,\ \eta = \rho = 1,$ $u_{ij}(k,1) = 1\ (i,j \in \mathscr{R}, k = 1,2,...,120),$ $\hat{\mathbf{\Phi}}(k,1) = \begin{bmatrix} -0.5 & 0.5 & -0.5 & 0.5 & 0 & 0 \\ 0.5 & -0.5 & 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0.5 & -0.5 & 0.5 & -0.5 \end{bmatrix}$	proposed method
P-type ILC	$\mathbf{u}(k,l) = K_{ILC} *$ $(\mathbf{n}_{crit} - \mathbf{n}(k+1, l-1))$	$K_{ILC} = 10^{-5} \times \begin{bmatrix} -6.6 & 6.0 & -4.8 & 6.6 & 0 & 0 \\ 4.8 & -7.2 & 0 & 0 & -4.2 & 6.0 \\ 0 & 0 & 6.0 & -7.2 & 6.6 & -7.2 \end{bmatrix}^T$	compare with the adaptive feature of MFAILPC
PI controller	$\mathbf{u}(k) = K_P * (\mathbf{n}_{crit} - \mathbf{n}(k))$ $+ K_I * \sum_{\nu=0}^{k} (\mathbf{n}_{crit} - \mathbf{n}(\nu))$	$K_P = 10^{-5} \times \begin{bmatrix} -2.1 & 3 & -2.1 & 3 & 0 & 0 \\ 3 & -2.4 & 0 & 0 & -2.1 & 3 \\ 0 & 0 & 3 & -2.4 & 3 & -2.4 \end{bmatrix}^T$ $K_I = 10^{-5} \times \begin{bmatrix} -1.9 & 2.2 & -1.8 & 2.4 & 0 & 0 \\ 2.4 & -1.7 & 0 & 0 & -0.7 & 2.4 \\ 0 & 0 & 2.3 & -1.9 & 0.9 & -1.9 \end{bmatrix}^T$	compare with the learning feature of MFAILPC
MPC	perimeter control-only formulation in eq. (10) in Sirmatel and Geroliminis (2018)	$N_p = 25,\ \Delta_u = 0.2,$ dynamics (1)-(8) with all network parameters	compare with the data-driven feature of MFAILPC
DED-MFAC	eq.(24), (28) and (29) in Lei et al. (2019)	$\zeta = 5000,\ \lambda_i = 9.1,\ \mu = 0.8$ $\eta = 1.5,\ \rho = 1,$	previous work using MFAC framework



(a) (b)

Figure 4: MFAILPC parameter selection test: (a) TTS over iterations with different λ; (b) TTS over iterations with different μ.

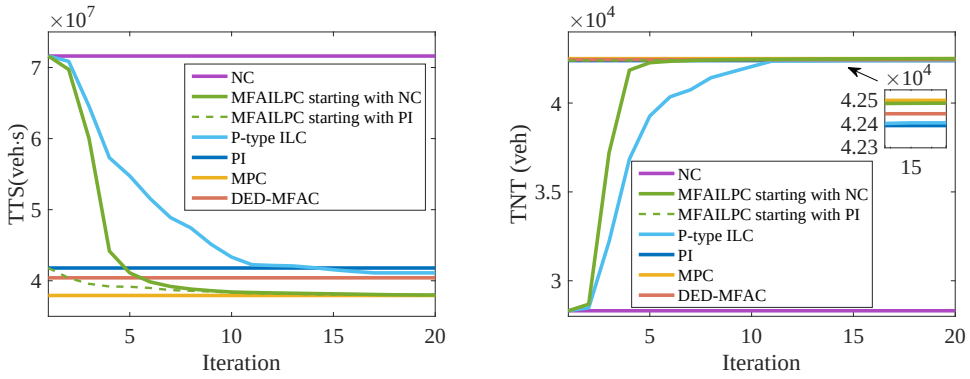


Figure 5: Results in Case I: (a) TTS over iterations; (b) TNT over iterations

4.2. Controller Parameter Selections

Parameters of the controllers are listed in Table 2. For MFAILPC, λ and μ are two main parameters. Therefore, two parameter sensitivity tests are also conducted with different values of λ and μ, which is shown in Fig. 4. From the test, one can find that the values of λ and μ mainly affect the convergence speed and have less effect on the final TTS. As discussed in Remark 3, this is mainly because MFAILPC adaptively tunes the learning gain during the iteration process. In terms of control performance and the smoothness of inputs, we choose $\lambda = 0.5$ and $\mu = 0.01$ in this paper.

4.3. Simulation results

1) Case I: No Errors

In this case, no error is considered in the plant. For MFAILPC, P-type ILC and PI control, the critical accumulation $\mathbf{n}_{crit}$ used in the algorithm is consistent with the one in the plant. For MPC, the predictive model is

assumed to be exactly the same with the plant.

The results are shown in Fig. 5-9. The summary of TTS and TNT over iterations is presented in Fig. 5. In Fig. 5(a), it can be seen the TTS in the NC case is very high because of the congestion, while the TTS has a significant drop with perimeter control. Since the ILC strategy learns from the history information, the TTS in MFAILPC and P-type ILC decrease with the iterations. It shows that MFAILPC performs better than PI control and DED-MFAC when the iteration is larger than 5 and 6, respectively. And it will finally reach a TTS almost as good as the MPC. Although MPC performs very well in this ideal case, it should pointed out that MPC needs more detailed data like $d_{ij}(t)$ and its performance depends on the accuracy of the model. As for the P-type ILC, it can only get a TTS slightly less than the PI scheme for the 20^{th} iteration. The performance in terms of TNT presents similar trends for each strategy as shown in Fig. 5(b). In practical implementation, MFAILPC can improve the performance by selecting the initial control inputs from other existing perimeter strategies. (Please see more detailed discussions on this issue in Hou et al. (2011)) The green dash curves indicate that MFAILPC will work much better in the beginning iterations if it initializes the control input form a well-tune PI controller. The evolution of accumulations and perimeter inputs in the 20^{th} iteration are given in Fig. 6-7. In the NC case, region 1 becomes gridlocked due to high demands. On the contrary, the transfer flows are balanced among all regions and region 1 is no longer congested under perimeter control.

To further study the iterative learning feature of MFAILPC, the time series of MFAILPC with 1-20^{th} iterations are compared with PI controller in

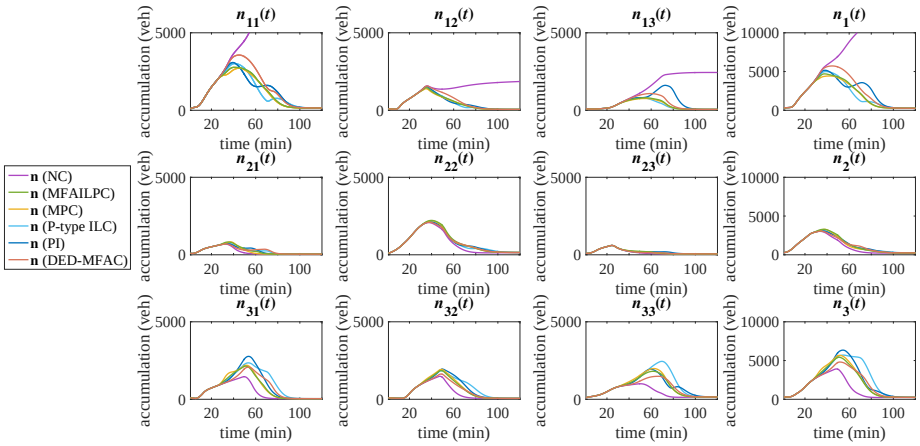


Figure 6: Time series of accumulations in the 20^{th} iteration under Case I.

Fig. 8-9. Here, the evolution of $u_{21}(t)$ is taken as an example to explain why MFAILPC performs better. In the 1-7^{th} iteration, $u_{21}(t)$ of MFAILPC has a trend similar to the PI controller. However, after the 7^{th} iteration, $u_{21}(t)$ decreases in an earlier time because historical data shows the control input is not restrictive enough for this period of time (see from 25 min to 60 min). Therefore, the accumulation under MFAILPC scheme is more likely to be kept around critical value, which leads to a higher trip completion rate and better traffic conditions overall.

2) Case II: Errors in MFDs

In case II, the MFDs used in the control schemes are scaled up or down, which is described in Fig. 10(a). In other words, the $\mathbf{n}_{crit}$ used in MFAILPC, P-type ILC and PI controller are deviated from the real value by percentages, while the predictive model utilized in MPC also has mismatch with the real plant according to this error.

Fig. 11(a) shows the TTS results with different MFD scale errors. The

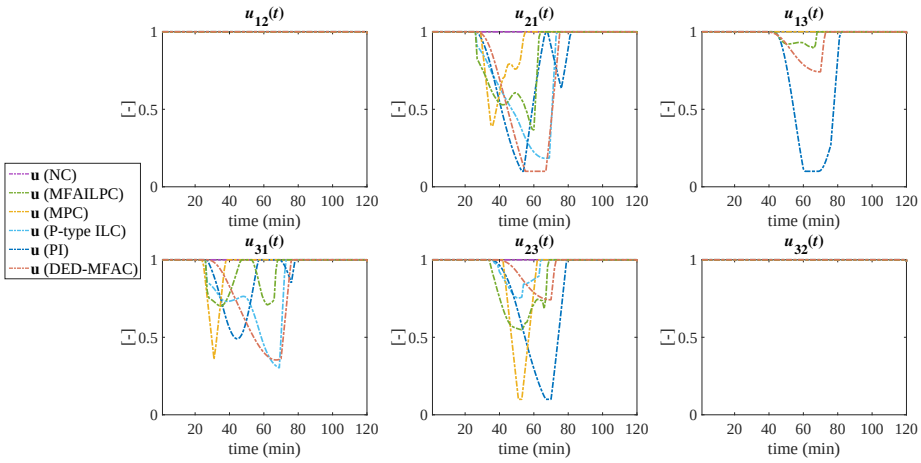


Figure 7: Time series of inputs in the 20^{th} iteration under Case I.

green line depicts the TTS of MFAILPC in the 20^{th} iteration. Comparing the green line with others, one will find that MFAILPC is more resilient than other methods since its TTS is always kept to a low level. It can be observed that MFAILPC can be the best option if the percentage of the scale errors is less than -4% or larger than 20%. The TTS of P-type ILC, DED-MFAC and PI control are likely to rise slightly with the increase of $\mathbf{n}_{crit}$. For the model-based schemes, MPC presents a satisfying performance in a wide range of scaling errors. However, the performance of MPC will deteriorate if the mismatch between the predictive model and the plant is high. For instance, if the scale of the MFD in the model is decreased by -10%, MPC will get even a higher TTS than PI control. In Fig. 11(b), a similar conclusion can also be obtained for TNT of each strategy.

3) Case III: Measurement Noise

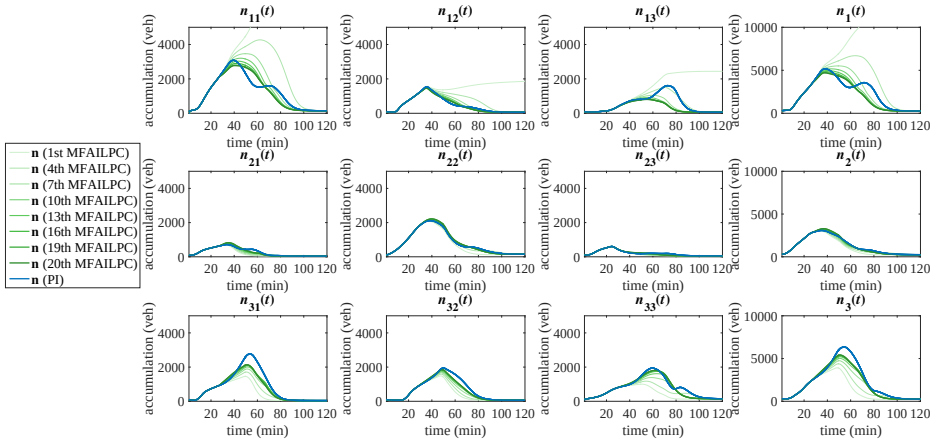


Figure 8: Accumulation comparison between PI control and MFAILPC in different iterations.

The following type of normal distribution measurement noise is tested.

$$\tilde{n}_{ij}(k) = n_{ij}(k) + \mathbb{N}(0, \sigma^2)$$

where $\tilde{n}_{ij}(k)$ is the measured accumulation in region i with destination region j and σ is the variance for the measurement noise. This type of noise may exist in the sensor. For instance, the car positioning system may fail to identify some vehicles in the network. An illustrative diagram of the measured accumulation ($\sigma = 40$) is shown in Fig. 10(b).

The TTS and TNT results with different level of measurement noise are plotted in Fig. 12. The curves in Fig. 12 reveal that both MFAILPC and MPC are robust to the measurement noise, and MFAILPC performs slightly better under the measurement noise if the the variance σ is high. P-type ILC , DED-MFAC and PI control also show acceptable results in both TTS and TNT but not as good as MFAILPC.

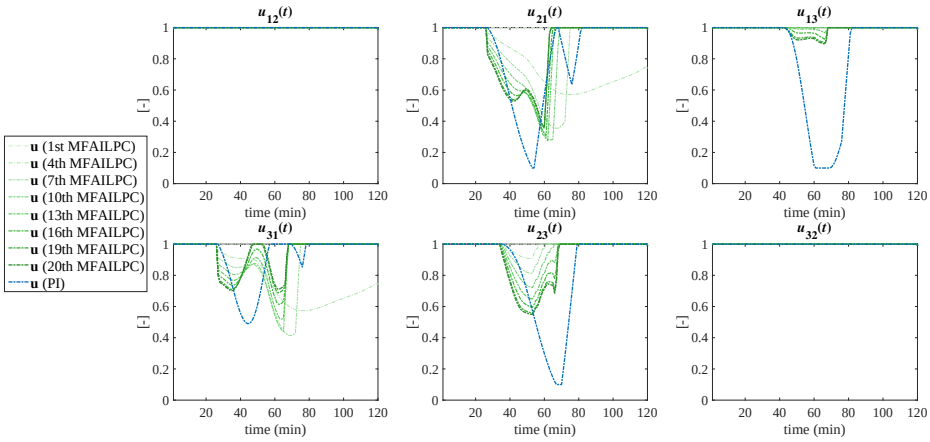
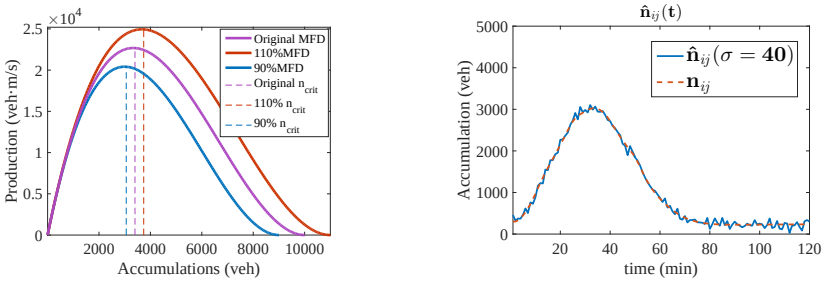


Figure 9: Input comparison between PI control and MFAILPC in different iterations



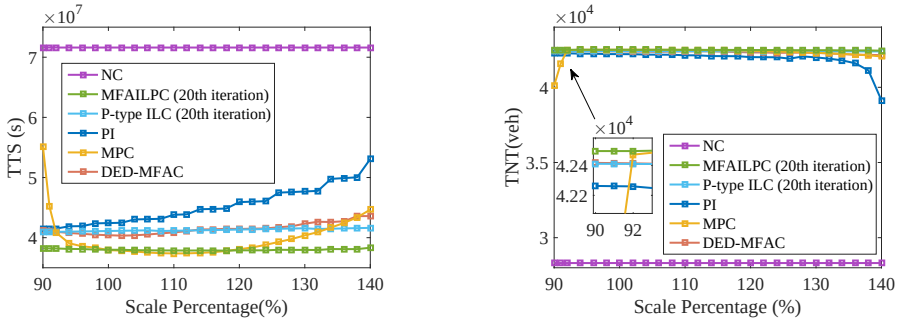
(a) (b)

Figure 10: Error illustration: (a) MFDs with different scales; (b) Measurement noise

4) Case IV: Iteration-Varying Demands

In Case IV, the demand is assumed to be iteration-varying in the simulation, i.e., the shape of the demand will change during different days. The iteration-varying demand profiles is shown in Fig. 13. Note that these profiles contain different types of demand changes including increase or decrease in the demand magnitude, time durations, and so on.

For the time-varying demand, TSPV and NTP are used to evaluate the



(a) (b)

Figure 11: Results in Case II: (a) TTS with different MFD scales; (b) TNT with different MFD scales

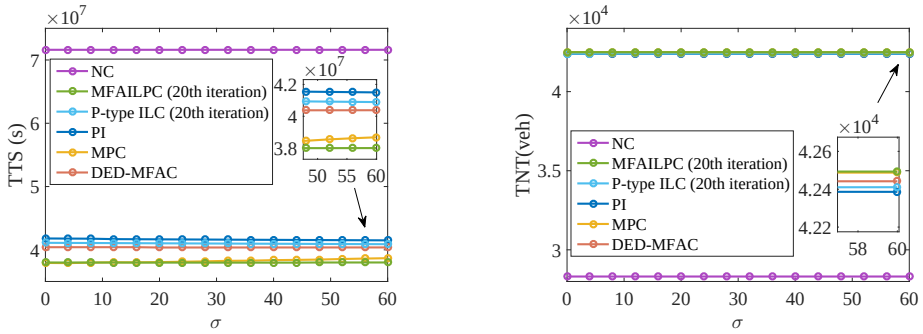
performance for different total demands, and the results are shown in Fig. 14. After some iterations of learning, MFAILPC performs almost as good as MPC, which means MFAILPC can also fulfill the task under a non-strict repetitive traffic flow pattern. In addition, MFAILPC does not need any information of the demand or plant model, while they are required for MPC.

5) Case V: Mixed Errors

A special case is examined here with 92% MFD scale error, measurement noise ($\sigma = 40$), and iteration-varying demand profiles. The TSPV and NTP change over iterations are shown in Fig. 15. After 5 iterations of learning, MFAILPC exceeds all other strategies in TSPV and NTP. In Fig. 15, the green dashed line presents the performance of MFAILPC starting with the perimeter inputs from the PI controller. It can be observed that MFAILPC greatly improves the performance in the first several iterations. And it only takes 2 times of iterations to get a fairly good result.

6) Case VI: Time-changing MFDs

In above simulations, the MFDs in the plant are assumed to be fixed.



(a) (b)

Figure 12: Results in Case III: (a) TTS with different levels of measurement noise; (b) TNT with different levels of measurement noise

However, due to the hysteresis phenomena(Geroliminis and Sun, 2011; Gayah and Daganzo, 2011; Mahmassani et al., 2013) and other factors, the MFD may also change over time. In this Case, a time-changing MFD is simulated in the network as shown in Fig. 16. Note that MFD2 used during $41 \leq t \leq 80$ is the same with the one in Case I-V and it has higher capacity than MFD1 and MFD3. The results are presented in Fig. 17. It can be noted, when dealing with the time-changing MFD, MFAILPC and P-type ILC outperform other strategies because of the learning ability. And MFAILPC is better than P-type ILC in terms of convergence speed and final performance. From a control point of view, the property of a repetitive process can be regarded as invariant in the iteration domain, even though it is varying in the time domain. On the contrary, the inaccurate model gives MPC inappropriate decisions, which decrease the performance facing the time-changing MFDs.

In summary, the simulation results and CUP time per step are listed in Table 3, where "Final" stands for the result in the 20^{th} iteration. The best

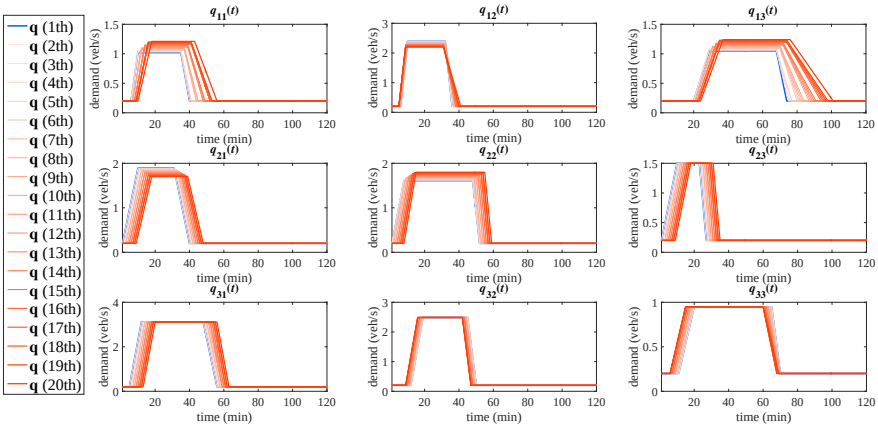
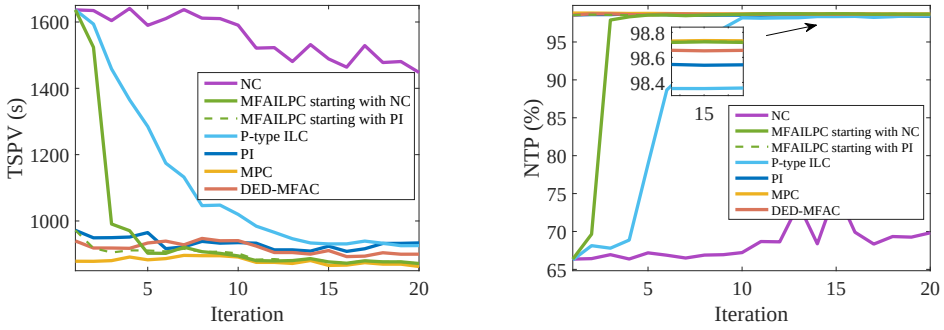


Figure 13: Iteration-varying demand profiles

result in each case is written in bold font. It can be seen the performance of MFAILPC is only second to MPC in Case I and IV, where MPC knows full information of the plant. When uncertainties are involved, MFAILPC is the most robust among other strategies as shown in Case II, III, V, and VI. In addition, the proposed method is more applicable since it do not needs model information or detailed data like demand profiles $d_{ij}(t)$. The computation time of MFAILPC is negligible with respect to the sampling time and thus can be applied in practice without any problems related to CPU efforts.Some disadvantages of MFAILPC should also be pointed here. First, the method is not recommended to used if the traffic flow repetitiveness is severely violated in some iterations, e.g., an accidental network-level change of traffic topology due to an emergency situation. In such a situation, the strategy should be switched to DED-MFAC or a PI controller (More detailed discussions on this issue can be found in Hou et al. (2011)). Second, MFAILPC may fail to perform better than other model-based methods if the predictive model is



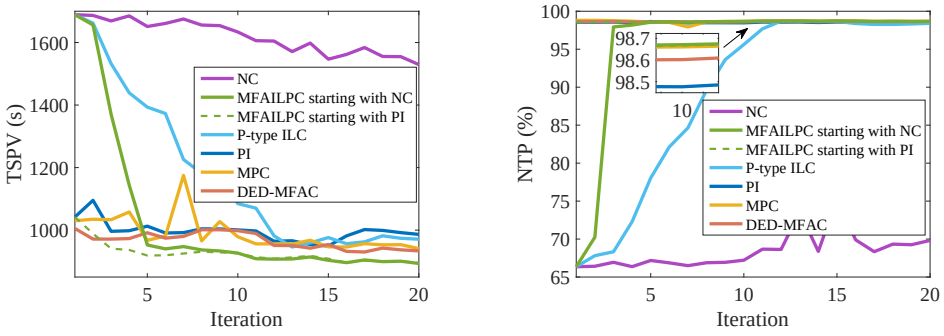
(a) (b)

Figure 14: Results in Case IV: (a) TSPV with iteration-varying demand; (b) NTP with iteration-varying demand

accurate and there is no extra uncertainty like measurement noise.

Table 3: The Performance Summary under Different Cases

Strategy	Case I	Case II	Case III	Case IV	Case V	Case VI	Avg. CPU Time (s)
	Final TTS (veh · s · 10^7)	Avg. TTS (veh · s · 10^7)	Avg. TTS (veh · s · 10^7)	Final TSPV (s · 10^2)	Final TSPV (s · 10^2)	Final TTS (veh · s · 10^7)	
NC	7.161	7.161	7.161	15.295	15.295	7.693	-
MFAILPC	3.799	**3.801**	**3.799**	8.721	**8.928**	**4.649**	0.0095
P-type ILC	4.111	4.127	4.099	9.258	9.710	4.729	0.0090
PI	4.179	4.512	4.163	9.338	9.864	5.572	**0.0061**
MPC	**3.793**	4.001	3.824	**8.627**	9.409	5.336	0.9700
DED-MFAC	4.041	4.146	4.039	8.998	9.335	5.025	0.0072
	Final TNT (veh · 10^4)	Avg. TNT (veh · 10^4)	Avg. TNT (veh · 10^4)	Final NTP (%)	Final NTP (%)	Final TNT (veh · 10^4)	Max. CPU Time (s)
NC	2.831	2.831	2.831	69.82	69.82	2.997	-
MFAILPC	4.250	**4.248**	**4.250**	98.67	**98.67**	**4.251**	0.0102
P-type ILC	4.241	4.240	4.240	98.47	98.41	4.249	0.0093
PI	4.239	4.194	4.239	96.15	98.48	4.227	**0.0065**
MPC	**4.251**	4.228	4.249	**98.69**	98.65	4.241	1.0788
DED-MFAC	4.244	4.233	4.244	98.62	98.67	4.243	0.0083



(a) (b)

Figure 15: Results in Case V: (a) TSPV over iterations; (b) NTP over iterations

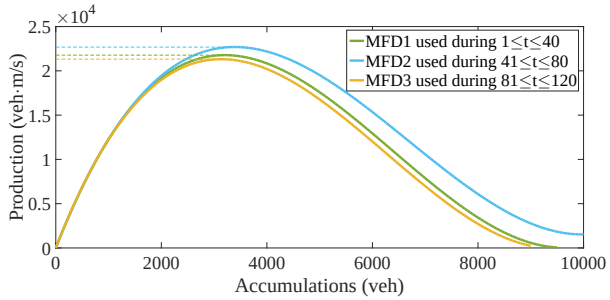
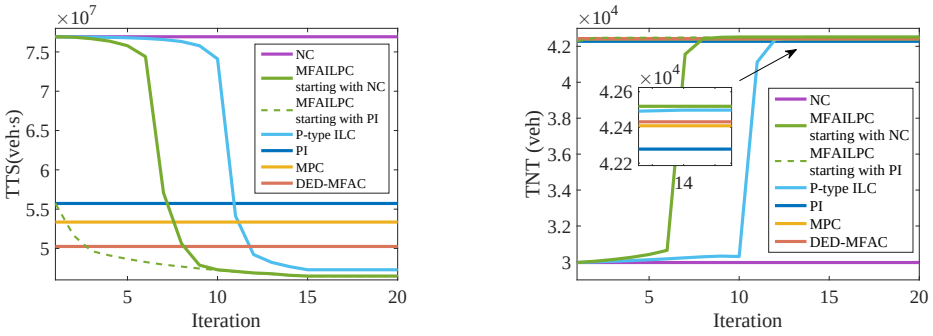


Figure 16: Time-changing MFDs

5. Conclusions

In this paper, the perimeter control problem for a multi-region network is addressed by a novel data-driven method named MFAILPC. By dynamically linearizing the controlled plant at each operation point, the system has been transformed into an iterative CFDL data model between successive iterations. With the help of the I/O data, the PJM is estimated and used in the controller design. There are three important features of MFAILPC: First, only the I/O data of the system is required in the algorithm and no predic-



(a) (b)

Figure 17: Results in Case VI: (a) TTS over iterations; (b) TNT over iterations

tive model or demand information are needed for the control computation. Second, the iterative learning feature is combined into the proposed method which means the controller will improve its performance from historical data. Third, the iterative learning gain is adaptively tuned with recently received data which endows the controller with stronger learning ability. From the comprehensive simulations tests conducted on a three-region MFDs network model, MFAILPC is verified to compete MPC in the circumstance without uncertainty and is more resilient than all other strategies considering modeling errors, measurement noise, demand fluctuations, and time-varying MFDs.

Future work could include the design of a modified iterative learning method that can directly minimize the TTS. Furthermore, real-world implementation of perimeter control is a research priority.

Acknowledgment

The first and second authors would like to acknowledge the support by the National Natural Science Foundation of China (Grant # 61433002, 61833001) and the Beijing Natural Science Foundation (Grant # L161007). The third and fourth authors acknowledge the support by the ERC (European Research Council) Starting Grant "METAFERW: Modelling and controlling traffic congestion and propagation in large-scale urban multimodal networks" (Grant # 338205).

Appendix A. The Proof of Theorem 1

From the definition $\Delta\mathbf{n}(k+1,l)$ and (9), one has

$$\begin{aligned}\Delta\mathbf{n}(k+1,l) &= \mathbf{f}(\mathbf{n}(k,l),\mathbf{u}(k,l),\mathbf{d}(k,l)) - \mathbf{f}(\mathbf{n}(k,l-1),\mathbf{u}(k,l-1),\mathbf{d}(k,l-1)) \\ &= \mathbf{f}(\mathbf{n}(k,l),\mathbf{u}(k,l),\mathbf{d}(k,l)) - \mathbf{f}(\mathbf{n}(k,l),\mathbf{u}(k,l-1),\mathbf{d}(k,l)) \\ &+ \mathbf{f}(\mathbf{n}(k,l),\mathbf{u}(k,l-1),\mathbf{d}(k,l)) \\ &- \mathbf{f}(\mathbf{n}(k,l-1),\mathbf{u}(k,l-1),\mathbf{d}(k,l-1)) \end{aligned} \tag{A.1}$$

Denote

$$\begin{aligned}\boldsymbol{\tau}(k,l) &= \mathbf{f}(\mathbf{n}(k,l),\mathbf{u}(k,l-1),\mathbf{d}(k,l)) \\ &\quad - \mathbf{f}(\mathbf{n}(k,l-1),\mathbf{u}(k,l-1),\mathbf{d}(k,l-1))\end{aligned} \tag{A.2}$$

According to the differential mean value theorem and Assumption 1, eq. (A.1) can be rewritten as

$$\Delta\mathbf{n}(k+1,l) = \frac{\partial\mathbf{f}^*}{\partial\mathbf{u}(k,l)}(\mathbf{u}(k,l)-\mathbf{u}(k,l-1)) + \boldsymbol{\tau}(k,l) \tag{A.3}$$

where $(\partial \mathbf{f}^*)/(\partial \mathbf{u}(k,l))$ denotes the partial derivative value of $\mathbf{f}(\cdot)$ with respect to $\mathbf{u}(k,l)$) at a certain point between

$$[\mathbf{n}(k,l), \mathbf{u}(k,l), \mathbf{d}(k,l)]^T$$

and

$$[\mathbf{n}(k,l), \mathbf{u}(k,l-1), \mathbf{d}(k,l)]^T$$

For every fixed time instant k and fixed iteration l, consider the following equation with a variable $\boldsymbol{\beta}(k,l)$:

$$\boldsymbol{\tau}(k,l) = \boldsymbol{\beta}(k,l)\Delta\mathbf{u}(k,l) \tag{A.4}$$

Since $\| \mathbf{u}(k,l)\| \neq 0$, there must exist at least one solution $\boldsymbol{\beta}^*(k,l)$ to eq. (A.4).

Let $\boldsymbol{\Phi}(k,l) = (\partial \mathbf{f}^*)/(\partial \mathbf{u}(k,l)) + \boldsymbol{\beta}^*(k,l)$. Then, eq. (A.3) is rewritten as $\Delta\mathbf{n}(k+1,l) = \boldsymbol{\Phi}(k,l)\Delta\mathbf{u}(k,l)$, which is the CFDL model in Theorem 1. The boundedness of $\boldsymbol{\Phi}(k,l)$ is guaranteed directly by using Assumption 2.

References

References

Aalipour, A., Kebriaei, H., Ramezani, M., 2018. Analytical optimal solution of perimeter traffic flow control based on mfd dynamics: A pontryagin's maximum principle approach. IEEE Trans. Intell. Transp. Syst., doi: 10.1109/TITS.2018.2873104.

Aboudolas, K., Geroliminis, N., 2013. Perimeter and boundary flow control in multi-reservoir heterogeneous networks. Transp. Res. Part B 55, 265–281.

Amann, N., Owens, D. H., Rogers, E., 1996. Iterative learning control for discrete-time systems with exponential rate of convergence. IEE Proceedings - Control Theory and Applications 143 (2), 217–224.

Ampountolas, K., Zheng, N., Geroliminis, N., 2017. Macroscopic modelling and robust control of bi-modal multi-region urban road networks. Transp. Res. Part B 104, 616–637.

Arimoto, S., Kawamura, S., Miyazaki, F., 1984. Bettering operation of robots by learning. Journal of Robotic Systems 1 (2), 123–140.

Ben-Akiva, M., Bierlaire, M., 1999. Discrete Choice Methods and their Applications to Short Term Travel Decisions. Springer.

Bristow, D. A., Tharayil, M., Alleyne, A. G., 2006. A survey of iterative learning control. IEEE Contr. Syst. Mag. 26 (3), 96–114.

Bu, X., Yu, Q., Hou, Z., Qian, W., 2019. Model free adaptive iterative learning consensus tracking control for a class of nonlinear multiagent systems. IEEE Trans. Syst., Man Cybern. 49 (4), 677–686.

Casadei, G., Bertrand, V., Gouin, B., Canudas-de Wit, C., 2018. Aggregation and travel time calculation over large scale traffic networks: An empiric study on the grenoble city. Transp. Res. Part C 95, 713–730.

Chen, L., Narendra, K. S., 2004. Identification and control of a nonlinear discrete-time system based on its linearization: a unified framework. IEEE Trans. Neural Netw. 15 (3), 663–673.

Chi, R. H., Hou, Z. S., 2007. Dual-stage optimal iterative learning control for nonlinear non-affine discrete-time systems. Acta Automatica Sinica 33 (10), 1061–1065.

Chi, R. H., Hou, Z. S., Huang, B., Jin, S. T., 2015. A unified data-driven design framework of optimality-based generalized iterative learning control. Comput. Chem. Eng. 77 (77), 10–23.

Daganzo, C. F., 2007. Urban gridlock: Macroscopic modeling and mitigation approaches. Transp. Res. Part B 41 (1), 49–62.

Diakaki, C., Papageorgiou, M., Aboudolas, K., 2002. A multivariable regulator approach to traffic-responsive network-wide signal control. Control Eng. Pract. 10 (1), 183–195.

Ding, H., Zhang, Y., Zheng, X., Yuan, H., Zhang, W., 2018. Hybrid perimeter control for two-region urban cities with different states. IEEE Trans. Control Syst. Technol. 26 (6), 2049–2062.

Fu, H., Liu, N., Hu, G., 2017. Hierarchical perimeter control with guaranteed stability for dynamically coupled heterogeneous urban traffic. Transp. Res. Part C 83, 18–38.

Gao, X., Gayah, V. V., 2018. An analytical framework to model uncertainty in urban network dynamics using macroscopic fundamental diagrams. Transp. Res. Part B 117, 660–675.

Gayah, V. V., Daganzo, C. F., 2011. Clockwise hysteresis loops in the macroscopic fundamental diagram: An effect of network instability. Transp. Res. Part B 45 (4), 643–655.

Geroliminis, N., Daganzo, C. F., 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. Transp. Res. Part B 42 (9), 759–770.

Geroliminis, N., Haddad, J., Ramezani, M., 2013. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. IEEE Trans. Intell. Transp. Syst. 14 (1), 348–359.

Geroliminis, N., Sun, J., 2011. Properties of a well-defined macroscopic fundamental diagram for urban traffic. Transp. Res. Part B 45 (3), 605–617.

Godfrey, J., 1969. The mechanism of a road network. Traffic Eng. Control 11 (7), 323–327.

Haddad, J., 2017a. Optimal coupled and decoupled perimeter control in one-region cities. Control Eng. Pract. 61, 134–148.

Haddad, J., 2017b. Optimal perimeter control synthesis for two urban regions with aggregate boundary queue dynamics. Transp. Res. Part B 96, 1–25.

Haddad, J., Geroliminis, N., 2012. On the stability of traffic perimeter control in two-region urban cities. Transp. Res. Part B 46 (9), 1159–1176.

Haddad, J., Mirkin, B., 2016. Adaptive perimeter traffic control of urban road networks based on mfd model with time delays. Int. J. Robust Nonlinear Control 26 (6), 1267–1285.

Haddad, J., Mirkin, B., 2017. Coordinated distributed adaptive perimeter control for large-scale urban road networks. Transp. Res. Part C 77, 495–515.

Haddad, J., Shraiber, A., 2014. Robust perimeter control design for an urban region. Transp. Res. Part B 68, 315–332.

Hou, Z., Xu, X., Yan, J., Xu, J., Xiong, G., 2011. A complementary modularized ramp metering approach based on iterative learning control and alinea. IEEE Trans. Intell. Transp. Syst. 12 (4), 1305–1318.

Hou, Z. S., Chi, R. H., Gao, H. J., 2017. An overview of dynamic linearization based data-driven control and applications. IEEE Trans. Ind. Electron. 64 (5), 4076 – 4090.

Hou, Z. S., Jin, S. T., 2013. Model Free Adaptive Control: Theory and Applications. CRC Press, Taylor and Francis Group, Florida.

Hou, Z. S., Li, X. Y., 2016. Repeatability and similarity of freeway traffic flow and long-term prediction under big data. IEEE Trans. Intell. Transp. Syst. 17 (6), 1786–1796.

Hou, Z. S., Wang, Z., 2013. From model-based control to data-driven control: Survey, classification and perspective. Information Sciences 235, 3–35.

Hou, Z. S., Xiong, S. S., 2019. On model free adaptive control and its stability analysis. IEEE Trans. Autom. Control, doi:10.1109/TAC.2019.2894586.

Hou, Z. S., Xu, J. X., 2003. Freeway traffic density control using iterative learning control approach. In: the IEEE 6th International Conference On Intelligent Transportation Systems. Shanghai, China, October 12-15, pp. 1081–1086.

Hou, Z. S., Xu, J. X., Yan, J., 2008. An iterative learning approach for density control of freeway traffic flow via ramp metering. Transp. Res. Part C 16 (1), 71–97.

Ji, Y., Geroliminis, N., 2012. On the spatial partitioning of urban transportation networks. Transp. Res. Part B 46 (10), 1639–1656.

Keyvan-Ekbatani, M., Kouvelas, A., Papamichail, I., Papageorgiou, M., 2012. Exploiting the fundamental diagram of urban networks for feedback-based gating. Transp. Res. Part B 46 (10), 1393–1403.

Keyvan-Ekbatani, M., Papageorgiou, M., Knoop, V. L., 2015a. Controller design for gating traffic control in presence of time-delay in urban road networks. Transp. Res. Part C 7 (6), 397–402.

Keyvan-Ekbatani, M., Yildirimoglu, M., Geroliminis, N., Papageorgiou, M., 2015b. Multiple concentric gating traffic control in large-scale urban networks. IEEE Trans. Intell. Transp. Syst. 16 (4), 2141–2154.

Kim, S., Tak, S., Yeo, H., 2018a. Agent-based network transmission model using the properties of macroscopic fundamental diagram. Transp. Res. Part C 93, 79–101.

Kim, S., Tak, S., Yeo, H., 2018b. Investigating transfer flow between urban networks based on a macroscopic fundamental diagram. Transp. Res. Rec. 2672 (20), 75–85.

Kouvelas, A., Saeedmanesh, M., Geroliminis, N., 2017. Enhancing model-based feedback perimeter control with data-driven online adaptive optimization. Transp. Res. Part B 96, 26–45.

Lei, T., Hou, Z. S., Ren, Y., 2019. Data-driven model free adaptive perimeter control for multi-region urban traffic networks with route choice. IEEE Trans. Intell. Transp. Syst., 10.1109/TITS.2019.2921381.

Liu, D., Yang, G. H., 2019. Data-driven adaptive sliding mode control of nonlinear discrete-time systems with prescribed performance. IEEE Trans. Syst., Man Cybern. 49 (12), 2598–2604.

Lopez, C., Krishnakumari, P., Leclercq, L., Chiabaut, N., Van Lint, H., 2017. Spatiotemporal partitioning of transportation network using travel time data. Transp. Res. Rec. 2623 (1), 98–107.

Lowrie, P. R., March 1982. Scats: The sydney co-ordinated adaptive traffic system - principles, methodology, algorithms. In: International Conference of Road Traffic Signalling. pp. 67–70.

Mahmassani, H. S., Saberi, M., Zockaie, A., 2013. Urban network gridlock: Theory, characteristics, and dynamics. Transp. Res. Part C 36, 480–497.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O., 2000. Constrained model predictive control: Stability and optimality. Automatica 36 (6), 789–814.

Menelaou, C., Kolios, P., Timotheou, S., Panayiotou, C., Polycarpou, M., 2017. Controlling road congestion via a low-complexity route reservation approach. Transp. Res. Part C 81, 118–136.

Mohajerpoor, R., Saberi, M., Vu, H. L., Garoni, T. M., Ramezani, M., 2019. H robust perimeter flow control in urban networks with partial information feedback. Transp. Res. Part B.

Ramezani, M., Haddad, J., Geroliminis, N., 2015. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. Transp. Res. Part B 74, 1–19.

Robertson, D. I., Bretherton, R. D., 1991. Optimizing networks of traffic signals in real time-the scoot method. IEEE Trans. Veh. Technol. 40 (1), 11–15.

Saab, S. S., 2003. Stochastic p-type/d-type iterative learning control algorithms. Int. J. Control 76 (2), 139–148.

Saeedmanesh, M., Geroliminis, N., 2016. Clustering of heterogeneous networks with directional flows based on "snake" similarities. Transp. Res. Part B 91, 250–269.

Saeedmanesh, M., Geroliminis, N., 2017. Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. Transp. Res. Part B 105, 193–211.

Sirmatel, I. I., Geroliminis, N., 2018. Economic model predictive control of large-scale urban road networks via perimeter control and regional route guidance. IEEE Trans. Intell. Transp. Syst. 19 (4), 1112–1121.

Tayebi, A., 2004. Adaptive iterative learning control for robot manipulators. Automatica 40 (7), 1195–1203.

Varaiya, P., 2013. Max pressure control of a network of signalized intersections. Transp. Res. Part C 36, 177–195.

Wu, D., Chen, H., Huang, Y., Chen, S., 2018. On-line monitoring and model-free adaptive control of weld penetration in vppaw based on extreme learning machine. IEEE Trans. Ind. Informat., doi:10.1109/TII.2018.2870933.

Xi, Y. G., Wang, F., Wu, G. H., 1996. Nonlinear multi-model predictive control. IFAC Proceedings Volumes 29 (1), 2359–2364.

Xu, J. X., 2011. A survey on iterative learning control for nonlinear systems. Int. J. Control 84 (7), 1275–1294.

Yang, K., Menendez, M., Zheng, N., 2019. Heterogeneity aware urban traffic control in a connected vehicle environment: A joint framework for congestion pricing and perimeter control. Transp. Res. Part C 105, 439–455.

Zhong, R., Chen, C., Huang, Y., Sumalee, A., Lam, W., Xu, D., 2017. Robust perimeter control for two urban regions with macroscopic fundamental diagrams: a control-lyapunov function approach. Transp. Res. Part B 23, 922–941.

Zhong, R. X., Huang, Y. P., Chen, C., Lam, W. H. K., Xu, D. B., Sumalee, A., 2018. Boundary conditions and behavior of the macroscopic fundamental diagram based network traffic dynamics: A control systems perspective. Transp. Res. Part B 111, 327–355.

Zhou, Z., Schutter, B. D., Lin, S., Xi, Y., 2017. Two-level hierarchical model-based predictive control for large-scale urban traffic networks. IEEE Trans. Control Syst. Technol. 25 (2), 496–508.