STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR0003 (REV 10/98)

ADA Notice

For individuals with sensory disabilities, this document is available in alternate formats. For information call (916) 654-6410 or TDD (916) 654-3880 or write Records and Forms Management, 1120 N Street, MS-89, Sacramento, CA 95814.

1. REPORT NUMBER CA17-2959	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER

4. TITLE AND SUBTITLE

Traffic Predictive Control: Case Study and Evaluation

5. REPORT DATE

June 26, 2017

6. PERFORMING ORGANIZATION CODE

7. AUTHOR

Samuel Coogan and Maxence Dutreix , UCLA and UCCONNECT

8. PERFORMING ORGANIZATION REPORT NO.

UCCONNECT

9. PERFORMING ORGANIZATION NAME AND ADDRESS

The Regents of the University of California, Berkeley
Sponsored Projects Office
2150 Shattuck Ave, Ste #300
Berkeley, CA 94704

10. WORK UNIT NUMBER

11. CONTRACT OR GRANT NUMBER

65A0529 TO 045 Task 2959

12. SPONSORING AGENCY AND ADDRESS

California Department of Transportation
Division of Procurements and Contracts
1737 30th St. MS#65
Sacramento, CA 95816

13. TYPE OF REPORT AND PERIOD COVERED

Final Report
May 1, 2016 to April 30, 2017

14. SPONSORING AGENCY CODE

Caltrans DRISI

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This project developed a quantile regression method for predicting future traffic flow at a signalized intersection by combining both historical and real-time data.

The algorithm exploits nonlinear correlations in historical measurements and efficiently solves a quantile loss optimization problem using the Alternating Direction Method of Multipliers (ADMM). The resulting parameter vectors allow determining a probability distribution of upcoming traffic flow. These predictions establish an efficient, delay-minimizing control policy for the intersection. The approach is demonstrated on a case study with two years of high resolution flow measurements. It is emphasized that the results are applicable to any traffic intersection equipped with sensors that provide sufficiently high resolution of data acquisition.

In particular, the data must have sufficient spatial resolution, e.g., measuring turning counts, and sufficient temporal resolution, e.g., measurements each 15 minutes. For example, numerous sites in California, including a large number of intersections in LA County, possess sensors that provide the required data to a central server.

17. KEY WORDS Model-based Traffic Control Design, Quantile Predictions/Estimating, Traffic Flow at Intersections, Algorithm, Reduce Traffic Delays	18. DISTRIBUTION STATEMENT UTC Contract with UCCONNECT with UCLA

19. SECURITY CLASSIFICATION *(of this report)* Unclassified	20. NUMBER OF PAGES 21	21. COST OF REPORT CHARGED

Reproduction of completed page authorized.

DISCLAIMER STATEMENT

This document is disseminated in the interest of information exchange. The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California or the Federal Highway Administration. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement by the Department of any product described herein.

For individuals with sensory disabilities, this document is available in alternate formats. For information, call (916) 654-8899, TTY 711, or write to California Department of Transportation, Division of Research, Innovation and System Information, MS-83, P.O. Box 942873, Sacramento, CA 94273-0001.

Traffic Predictive Control:

Case Study and Evaluation

PI: Samuel Coogan
Graduate Student Researcher: Maxence Dutreix
Caltrans Task Manager: Lee Provost

Abstract

This project developed a quantile regression method for predicting future traffic flow at a signalized intersection by combining both historical and real-time data. The algorithm exploits nonlinear correlations in historical measurements and efficiently solves a quantile loss optimization problem using the Alternating Direction Method of Multipliers (ADMM). The resulting parameter vectors allow determining a probability distribution of upcoming traffic flow. These predictions establish an efficient, delay-minimizing control policy for the intersection. The approach is demonstrated on a case study with two years of high resolution flow measurements. It is emphasized that the results are applicable to any traffic intersection equipped with sensors that provide sufficiently high resolution of data acquisition. In particular, the data must have sufficient spatial resolution, e.g., measuring turning counts, and sufficient temporal resolution, e.g., measurements each 15 minutes. For example, numerous sites in California, including a large number of intersections in LA County, possess sensors that provide the required data to a central server.

1 INTRODUCTION

Despite the emergence of high-resolution sensing technologies in transportation systems, many traffic control approaches used in practice still fail to adequately leverage real-time and historical measurements [1]. Current demographic and urbanization trends worldwide likely portend a global over congestion of roads in the coming years [2, 3], raising the need of more optimized signal timing practices. Although typical signalized intersections are often able to accommodate moderate deviations from average traffic conditions, they lack the ability to adapt to more significant and uncommon variations in vehicle flows. Harvested real-time data, analyzed in tandem with historical information, provide a practical solution to this problem, as they enable us to predict the future state of traffic and to modify the intersection's behavior accordingly.

Previous work has demonstrated the strong potential of prediction-based control in a variety of traffic settings. Tools such as ARMAX models or Kalman filtering have delivered promising results in the framework of freeway traffic predictions [4, 5]. However, most models rely on pointwise forecasting techniques and solely estimate a single (e.g. most likely) future traffic condition. In systems displaying a high degree of uncertainty, determining only the most probable outcome is often not adequate for the implementation of an effective and robust control strategy.

Another recent contribution exploits low-rank latent structure in historical traffic data to predict future flows at intersections [6]. Highly correlated, low-rank components are computed and used directly as linear regressors for the prediction target. However, this technique is still restricted to the framework of pointwise forecasting.

This report builds on [6] and extend its capabilities with the utilization of probability forecasting tools. In particular, the aim is to predict future traffic flow at signaled intersections and assign a probability of occurrence to several ranges of possible flow values. Also, an objective is to capture the nonlinear relationships between past and future traffic flows, and to exploit them in the procedure. As these predictions ultimately need to be coordinated with real-time measurements, this work designs a computationally inexpensive, time-efficient algorithm by means of *multiple quantile regression analysis*. Lastly, the report presents a direct application of these results in a delay minimizing control policy.

First, a dimensionality-reduction algorithm is presented that casts the flow measurements vectors onto a smaller set of highly correlated components. Inspired by [7], which seeks a quantile regression algorithm for wind and power forecasting, the next step is to project the reduced-size data to a nonlinear feature space through the application of *radial basis functions* (RBF). Finally, a quantile loss function minimization problem is solved in order to compute a set of regression parameters and predict the quantiles of future traffic flow using an input vector collected in real-time.

We demonstrate our results on a case study using an intersection in South Carolina because an extensive database of traffic flow measurements at this intersection was available to the researchers through an existing industry collaboration. However, it is emphasized that the results are applicable to any traffic intersection equipped with sensors that provide sufficiently high resolution of data acquisition. In particular, the data must have sufficient spatial resolution, e.g., measuring turning counts, and sufficient temporal resolution, e.g., measurements each 15 minutes. For example, LA County maintains a number of traffic intersections that are able to provide the required data.

This paper is organized as follows: Section II presents the problem formulation. Section III describes the methodology used to predict the quantiles of future traffic flow from a set of training input and output. Section IV demonstrates the practical benefits of the quantile regression algorithm on the test site in Fig.1.

2 PROBLEM FORMULATION

Consider a traffic intersection with several approaches, each permitting a fixed number of movements. The total number of allowed movements at the intersection is denoted by M. For example, the intersection in Fig.1 used for the study is a standard intersection located in Beaufort, South Carolina. It is composed of 4 different approachs: North Bound (NB), East Bound (EB), South Bound (SB) and West Bound (WB). Each approach allows 3 distinct movements: Through (T), Right Turn (RT) and Left Turn (LT). Thus, there are M = 12 movements total. As shown in Fig. 2, traffic flows for these movements vary widely around the mean from day to day, rendering average-based control suboptimal. The goal is to exploit historical data along with real-time measurements in order to predict future vehicle flows. Subsequently, these predictions are used to adjust the signal timing control and better accommodate the ensuing traffic conditions.

A probabilistic forecasting problem consists in determining the probability density function $\mathcal{P}(Y \mid X = x)$, of a target random variable $Y \in \mathbb{R}$, with $x \in \mathbb{R}^m$ denoting the prediction's covariates. In this work, x compiles past traffic flows up to a given time for all M movements and y designates flows at a future time for a specified movement. The goal is to characterize $\mathcal{P}$ by a set of q predicted quantiles $\{\tilde{y}^{(\alpha_1)}, \tilde{y}^{(\alpha_2)}, \ldots, \tilde{y}^{(\alpha_q)}\}$, with each $\tilde{y}^{(\alpha_i)} \in \mathbb{R}$, where $\tilde{y}^{(\alpha_i)}$ is the predicted α_i-quantile for some $\alpha_i \in [0, 1]$. The number $y^{(\alpha)} \in \mathbb{R}$ satisfying $p(Y \leq y^{(\alpha)}) = \alpha$ is called the α-quantile for the random

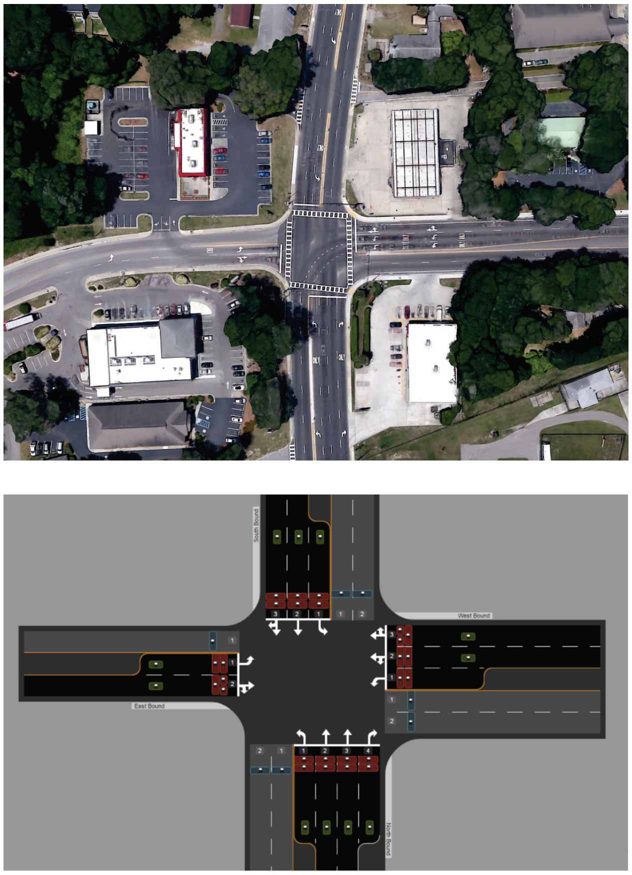

Figure 1: Aerial picture (top) and diagram (bottom) of the test site in Beaufort, South Carolina, displaying all four approaches and their associated movements. Car flow measurements are collected via stopbar, departure lane and advance sensors, depicted respectively as red, blue and green slabs [8].

variable Y.

For example, suppose that at time 10:00, the goal is to predict the total traffic flow for movement NB-T over the next hour, 10:00–11:00. Then, x contains traffic flows from 0:00 to 10:00 for all movements, y designates the total flow for movement NB-T between 10:00 and 11:00, and the set $\{\tilde{y}^{(\alpha_i)}\}_{i=1}^{q}$ contains q predicted quantiles for y. In addition, it is desired that the method works for an arbitrary number q of α-quantiles, $\alpha \in [0, 1]$.

First, an appropriate metric is established to gauge the quality of a quantile-based regression. Let $\rho_a(z)$ be the *tilted absolute loss function*, defined as $\rho_a(z) = \max\{az,\ (a-1)z\}$. Assume the set $S = \{y_1, y_2, \ldots, y_n\}$ to be a collection of random outcomes for y. The quantity

$$\sum_{i=1}^{q} \sum_{j=1}^{n} \rho_{\alpha_i}(\tilde{y}^{(\alpha_i)} - y_j) \tag{1}$$

is minimized by setting $\tilde{y}^{(\alpha_i)}$ as the true α_i-quantile of S [9]. Due to this property, the tilted absolute loss function is regarded as a standard measure of precision for quantile regressions and is the metric of quality used in this paper.

In Section III, using a set of n training input vectors $\{x_i\}_{i=1}^{n}$, with $x_i \in \mathbb{R}^m$, along with a set of training output scalars $\{y_i\}_{i=1}^{n}$, $y_i \in \mathbb{R}$, an efficient algorithm is developed for predicting the α_i-quantiles $\tilde{y}^{(\alpha_i)}$, $i = 1,\ 2,\ \ldots,\ q,$ as a function of an input vector $\hat{x} \in \mathbb{R}^m$ in order to minimize (1).

A case study is conducted in Section IV to illustrate the tools developed in the previous section using 2 years worth of traffic flow data collected at the intersection shown in Fig. 1.

3 QUANTILE REGRESSION

In the present setting, the aim is to exploit the fact that predictors and predictands correlate in a nonlinear fashion. To that end, the first step is to construct a nonlinear transformation $T : \mathbb{R}^m \to \mathbb{R}^k$ from input vector $x \in \mathbb{R}^m$ to a nonlinear feature vector $T(x) \in \mathbb{R}^k$. Then, the objective is to find a collection of estimation parameters $\{\theta_i\}_{i=1}^{q}$, with each $\theta_i \in \mathbb{R}^k$ and such that $\tilde{y}^{(\alpha_i)} = \theta_i^T T(x)$. Further, T is chosen as a composition $\phi \circ H$, where $H : \mathbb{R}^m \to \mathbb{R}^{m'}$ is a dimensionality reduction operator and $\phi : \mathbb{R}^{m'} \to \mathbb{R}^k$ nonlinearly transforms the lower-dimensionality predictors to the feature space. Below, H is constructed as a *Projection to Latent Structure* (PLS) mapping, also known as *Partial Least Squares*, and ϕ using radial basis functions.

First, the relevance of PLS mapping and radial basis functions to the present case is discussed. Afterwards, it is shown how the *Alternating Direction Method of Multipliers* (ADMM) is used for efficiently computing the set $\{\theta_i\}_{i=1}^{q}$.

3.1 PLS Dimensionality Reduction

In the context of traffic predictions, a substantial number of predictors must be considered. For instance, a 15-minute sample interval of vehicle flows results in daily mea-

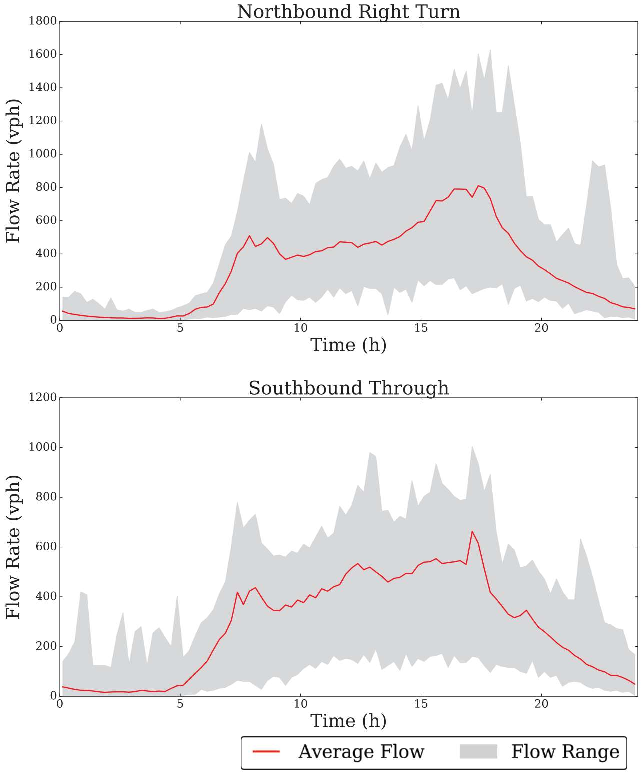

Figure 2: Examples of historical flow measurements for the Northbound Right Turn and Southbound Through movements. The red line indicates the average flow over the course of one day, the grey envelope shows the range of historical flow measurements throughout the day. Note that there is considerable variation around the mean.

surement vectors with length $4 \times 24 \times M$. This number rapidly grows inconvenient because nonlinear feature generation is a computationally costly and size-sensitive task. Dimensionality impracticalities arise, and it becomes difficult to disentangle the various relationships existing within the data set.

Here, it is proposed to first reduce the data size using the Projection to Latent Structure method [10]. Similar in some aspects to Principal Component Analysis, a PLS-based approach seeks to project the data onto a smaller set of orthogonal vectors in directions of high covariance between X and Y. The algorithm thus determines a low-rank representation of X, whose components are in turn greatly correlated with Y. This property also renders PLS a valuable tool to avoid overfitting. Denote by m' the number of PLS components to be computed, a user-specified tuning parameter.

Recall the set of training input vectors $\{x_i\}_{i=1}^{n}$ and outputs $\{y_i\}_{i=1}^{n}$. Let us introduce the mean-centered matrices $\tilde{X} \in \mathbb{R}^{n \times m}$ and $\tilde{Y} \in \mathbb{R}^{n}$ as

$$\tilde{X} = \begin{bmatrix} \tilde{x}_1^T \\ \vdots \\ \tilde{x}_n^T \end{bmatrix} \qquad \tilde{Y} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_n \end{bmatrix} , \tag{2}$$

where

$$\tilde{x}_i = x_i - \bar{x} \;\text{ with }\; \bar{x}_j = \frac{1}{n}\sum_{i=1}^{n}(x_i)_j , \tag{3}$$

$$\tilde{y}_i = y_i - \bar{y} \;\text{ with }\; \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i , \tag{4}$$

$(x_i)_j$ and $\bar{x}_j$ denote the jth entry of x_i and $\bar{x}$ respectively.

PLS analysis is carried out iteratively: a pair (p_i, s_i) of *principal component* p and *score vector* s is determined at each iteration of the process, with $p \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$. The contributions of newly computed components is removed by subtracting them from the data matrices between successive iterations. The algorithm terminates once m' component-score pairs have been calculated.

A classic algorithm for PLS is now briefly described [10]. In order to determine the pair (p_1, s_1), first find two solutions $v^* \in \mathbb{R}^n$ and $w^* \in \mathbb{R}^m$ solving the following optimization problem:

$$(v^*, w^*) = \underset{\|v\|_2=1\ ;\ \|w\|_2=1}{\text{argmax}} \quad [cov(\tilde{X}v, \tilde{Y}w)]^2 \tag{5}$$

$$= \underset{\|v\|_2=1\ ;\ \|w\|_2=1}{\text{argmax}} \quad (\tilde{X}v)^T(\tilde{Y}w) , \tag{6}$$

where *cov* denotes the covariance operator. This follows directly from the definition of latent structures evoked previously. Observe that v^* and w^* are, respectively, the left and right singular vectors of the product $\tilde{X}^T\tilde{Y}$, making this step computationally straightforward. Then, obtain a score vector s_1 by projecting $\tilde{X}$ onto the direction of high covariance v^* found in (6) and according to

$$s_1 = \frac{\tilde{X}v^*}{\|\tilde{X}v^*\|_2} \ . \tag{7}$$

Note that only v^* is involved in the computation of the score vector. This necessary asymmetry is introduced to later use the score vectors for regression.

Corresponding components p_1 and q_1 result from the projection of the data matrices onto s_1 and are given by

$$p_1 = \tilde{X}s_1 \ , \tag{8}$$

$$q_1 = \tilde{Y}s_1 \ . \tag{9}$$

Finally, update $\tilde{X}$ and $\tilde{Y}$ to generate $\tilde{X}_2$ and $\tilde{Y}_2$ according to

$$\tilde{X}_2 = \tilde{X} - s_1 p_1^T \ , \tag{10}$$

$$\tilde{Y}_2 = \tilde{Y} - s_1 q_1^T \ . \tag{11}$$

This process is repeated with the updated data matrices $\tilde{X}_2$, $\tilde{Y}_2$ and find an additional pair (p_2, s_2), etc. This algorithm is iterated to obtain m' pairs.

Once a collection $\{(p_i, s_i)\}_{i=1}^{m'}$ of component-score pairs has been computed, we build the reduced-size matrix of predictors S along with the loading matrix P and write

$$S = [s_1, \ s_2, \ \ldots, \ s_{m'}] \ , \tag{12}$$

$$P = [p_1, \ p_2, \ \ldots, \ p_{m'}] \ . \tag{13}$$

Denote by S_i the ith row of the PLS score matrix S. The PLS effectively fulfills its purpose of dimensionality reduction by representing $X \in \mathbb{R}^{n\times m}$, which contains raw traffic input data, as a score matrix $S^{n\times m'}$, $m' \ll m$, so that each training input$x_i \in \mathbb{R}^m$ is instead represented by the score vector $S_i \in \mathbb{R}^m$ whose components are maximally correlated with the set of output flows $\{y_i\}_{i=1}^n$.

Now, assume an input $\hat{x}$ is to be used for predictions. Its PLS projection score vector $\hat{S} \in \mathbb{R}^{m'}$ must be calculated with respect to the components of P. $\hat{S}$ is found using the expression

$$H(\hat{x}) := \hat{S} = ((\hat{x} - \bar{x})^T (P^T)^\dagger)^T \in \mathbb{R}^{m'}, \tag{14}$$

where $(P^T)^\dagger$ stands for the Moore-Penrose pseudoinverse of P^T.

3.2 Nonlinear Features Generation

Now that H has been characterized to reduce the data dimensionality, a nonlinear features transformation ϕ is defined. Among the most popular kernels employed in machine learning for nonlinear features extraction is the RBF Gaussian kernel [11]. Given a user-specified number of desired nonlinear features, the method finds a set of data centers and bandwidths used in the computation of the RBFs.

Initially, a k-means clustering algorithm is used on S to generate a set of k data centers $\mu_i \in \mathbb{R}^{m'}$ along with their associated bandwidth $\sigma_i \in \mathbb{R}$, such that $\sigma_i = \underset{l \neq i}{\text{median}} \; \|\mu_i - \mu_l\|_2; \; i = 1, \; 2, \ldots, \; k$. A multiple seeding *k-means*++ procedure is applied so as to ensure similar clustering for separate executions of the algorithm [12]. The RBF vector $\phi(U) = [\phi_1(U), \; \phi_2(U), \; \ldots, \; \phi_k(U)] \in \mathbb{R}^k$ is defined with

$$\phi_j(U) := e^{\frac{\|U - \mu_j\|_2}{2\sigma_j}}; \quad j = 1, \; 2, \; \ldots, \; k \tag{15}$$

being the RBF functions with center μ_j and bandwidth σ_j. The stacked matrix Φ of *feature vectors* $\phi(S_i) \in \mathbb{R}^k$ is constructed by evaluating the RBF vector for all S_i and concatenating them to obtain

$$\Phi = \begin{bmatrix} \phi(S_1) \\ \phi(S_2) \\ \vdots \\ \phi(S_n) \end{bmatrix} \in \mathbb{R}^{n \times k}. \tag{16}$$

Each row $\phi(S_i)$ of Φ is equivalent to the nonlinear transformation T applied to x_i, and thus $T(x_i) := H(\phi(x_i)) = \phi(S_i), \;\; i = 1, 2, \ldots, n$.

3.3 Alternating Direction Method of Multipliers Algorithm

Recall the set of parameters $\{\theta_i\}_{i=1}^q$ that is to be computed such that the α_i-th quantile $\tilde{y}^{(\alpha_i)}$ satisfies $\tilde{y}^{(\alpha_i)} = \theta_i^T T(\hat{x})$. Minimizing the absolute tilted loss function in order to find $\{\theta_i\}_{i=1}^q$ belongs to the class of convex optimization problems [13]. The equation (1) is now reformulated so as to include a l_2-regularization parameter λ and highlight the dependence of the predicted quantiles $\tilde{y}^{(\alpha_i)} = \theta_i^T T(x_j)$ on the training inputs x_j. The optimization problem becomes

$$\underset{\theta_i}{\text{argmin}} \sum_{i=1}^{q} \sum_{j=1}^{n} \rho_i(\theta_i^T T(x_j) - y_j) + \lambda \sum_{i=1}^{q} \frac{\|\theta_i\|_2^2}{2} \; . \tag{17}$$

This expression decouples along θ_i and thus is q independent optimization problems. The main advantage of the ADMM procedure is to provide an efficient solution to (17) through a simultaneous computation of all the estimators and avoid the redundancy of casting a solver for each θ_i consecutively, as suggested in [7] and [14]. This procedure is shown in the pseudo-code in Algorithm 1.

ADMM is an iterative process characterized by its step size δ. No stopping conditions are implemented; instead, a fixed number of iterations T is introduced as an additional tunable hyperparameter.

Upon completion, the program returns a matrix $\Theta \in \mathbb{R}^{k \times q}$ containing the desired estimators

$$\Theta = \begin{bmatrix} | & | & & | \\ \theta_1 & \theta_2 & \dots & \theta_q \\ | & | & & | \end{bmatrix} . \tag{18}$$

Although ADMM offers a clear advantage in terms of computational time and complexity, independent quantile regressions may be the source of possible mathematical aberrations, such as estimating two quantiles $\tilde{y}^{(\alpha_a)}$ and $\tilde{y}^{(\alpha_b)}$, with $\tilde{y}^{(\alpha_a)} < \tilde{y}^{(\alpha_b)}$ when $\alpha_a > \alpha_b$. Some quantile regression approaches include additional constraints in order to prevent this, however, this would eliminate the computational efficiency of the ADMM approach. Instead,it was found that sorting the set of computed quantiles in increasing order is a reasonable way to fix this issue, as is done in [7].

4 CASE STUDY

4.1 Traffic Flow Prediction

The algorithm presented in Section III is now demonstrated using data collected at the test site in Fig.1 on weekdays from March 2014 to September 2016. This is $n = 591$ days worth of traffic flow measurements for each movement.

Vehicle counts for all movements were sampled on 15-minute intervals. Measurement times remain the same for all days across the data set. To generate the training data, we aggregate the measurements by calendar days into the set $\{x_i\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^{(12 \times 4 \times T_S)}$ is a row-vector containing all flows for all movements from 00:00 to T_S in 15-minute intervals on day i. The objective is to forecast a set of flow percentiles $\{\tilde{y}^{(0.01)},\ \tilde{y}^{(0.02)},\ \tilde{y}^{(0.03)}, \dots,\ \tilde{y}^{(0.99)}\}$ for a specified movement, at time T_P, on a given day, $T_P > T_S$. The training set of outputs $\{y_i\}_{i=1}^{n}$ thus contains flow measurements for that particular movement at time T_P for all days in the data set.

For this case study, the goal is to make hourly predictions for all movements based on historical flows, and therefore let T_S vary from 10:00 to 23:00 in one-hour increments on different days. At each time step, the target quantity for prediction is the hourly flow for all movements in the time range $[T_S;\ T_S + 1$ hour]. Algorithm 2 depicts the procedure used to make predictions for a given T_S. PLS and ADMM hyperparameters were tuned empirically and the following combination was found to yield consistent, high-quality predictions: $m' = 7$, $\lambda = 0.00022,\ k = 250,\ T = 150,$ $\delta = 0.5$. Subtracting the mean flow from the training and test outputs mitigates the effects of traffic volume on the ADMM optimization and allows for a uniform selection of parameters across the whole data set. The average flow is later re-added to the estimated quantiles.

The performance of the regression algorithm is further enhanced by taking additional predictors into consideration. Weather data such as temperature and precipitations are good candidates, as they capture underlying seasonal trends and potential poor driving conditions. Hourly precipitation and hourly average temperature obtained from 0:00 to time T_S are appended to the inputs $\{x_i\}_{i=1}^{n}$ of traffic flow measurements before each prediction. It was found that including weather conditions had a positive, although slight, overall impact on the prediction quality, which was able to foresee lower traffic activity on extremely cold days or in the event of moderate precipitation.

Fig. 3 displays the 10th to 90th percentile range predicted by the algorithm for high-volume movements, NB-RT, on 3 distinct test days. Both the 30th to 70th and 40th to 60th percentile ranges are delineated with darker blue tones. The observed flow, as well as the average flow across the entire data set, are superimposed on the plots as a cyan solid line and a red dotted line respectively. The days shown in the figure exhibit drastically dissimilar traffic conditions: July 2nd 2015, which fell right before a long weekend holiday and experienced higher-than-average traffic; February 24th 2015, a day with lower-than-average traffic due to winter weather; and January 1st 2015, a nationwide US holiday causing very unusual traffic. In the figure, it is seen that the algorithm successfully detects variations from average conditions and accurately predicts impending traffic flow. Not only do the quantiles capture the general traffic trend in all the examples, but they are also very coherent in a statistical sense. Indeed, a proper fraction of observed values — which should amount to about 20% — fall outside of the 10 to 90 percentile range.

Cumulative distribution functions (CDF) can be extrapolated from the sets of predicted quantiles. CDFs reveal informative visual clues on the probability density functions inferred by the quantiles. Two examples produced at peak traffic times are presented in Fig. 4. For higher volume predictions, e.g. on July 2, the CDF's graphs appear more linear than in the case of lower traffic volume. Therefore, the algorithm displays greater certainty for lower-than-average traffic predictions, since their CDFs allocate more probability mass to specific flow ranges. For above-average traffic, the predictions are conservative and predict more uniform traffic distributions. In addition, observed values represented by green vertical lines on the plots frequently lie in the vicinity of the estimated most probable flow.

In order to quantify the precision of our forecasts, the tilted loss in (1) is used to assign a prediction score to each day. The set $\{\tilde{y}^{(0.10)},\ \tilde{y}^{(0.30)},\ \tilde{y}^{(0.50)},\ \tilde{y}^{(0.70)},\ \tilde{y}^{(0.90)}\}$ of predicted quantiles is used to compute these scores. Although this score cannot assess the quality of the algorithm in absolute terms, it allows us to make direct comparisons between different quantile forecasts. On the same days previously studied, we evaluate the sum of the prediction scores for all movements from 10:00 to 23:00. As shown in Table I, our quantiles clearly outperform those extracted from a direct percentile computation over the historical data set. The average tilted loss score for the data set decreases from 60×10^2 with historical quantiles to 38×10^2 when using predicted quantiles. It is observed that historical quantiles are particularly unsuited when traffic flow deviates from the average conditions. For example, on January 1, 2015, historical quantiles yield a high tilted loss score of 444×10^2 whereas predicted quantiles receive a score of 35×10^2.

4.2 Delay-Optimizing Control using Predictions

To evaluate the practical benefits of the traffic prediction algorithm, it is considered to use the predictions to adjust control actions at the intersection. Typically, a traffic intersection controller supposes fixed arriving flow for each movement and optimizes *green splits*, that is, the fractions of time each movement is given a green signal to allow traffic flow [15]. Here, the benefits of adjusting these control parameters every hour is studied.

The Synchro software is a software package used extensively by traffic engineers to compute optimal control parameters at intersections. In particular, it employs a quantile-based approach for estimating delay for signalized intersections [16]. It assumes five different traffic arrival scenarios, generates the optimal green times and cycle time for each one of them, and finally averages the five delays computed with a simple equation. In the event that no green splits can accommodate one or several of the potential situations, the ”infeasible” cases are discarded in the delay computation. However, these scenarios — namely the 90, 70, 50, 30 and 10 flow percentiles — are approximated in the Synchro software presuming a Poisson-distributed arrival of vehicles with a nominal average arrival rate and do not reflect the actual behavior of the intersection.

Inspired by this Synchro percentile method, we first predict the quantiles

$$\{\tilde{y}_i^{(0.10)},\ \tilde{y}_i^{(0.30)},\ \tilde{y}_i^{(0.50)},\ \tilde{y}_i^{(0.70)},\ \tilde{y}_i^{(0.90)}\}$$

of future flows for all 12 movements each hour, with $\tilde{y}_i^{(\alpha_j)}$ denoting the α_j-quantile for movement i. Then, we aim to minimize delay given by *Webster's delay formula*

$$d_i = \frac{0.5C(1-\frac{g_i}{C})^2}{1-[X_i\frac{g_i}{C}]} + 900\left[(X_i-1)+\sqrt{(X_i-1)^2+\frac{4X_i}{s_i}}\right] \tag{19}$$

as defined in the Highway-Capacity Manual [17] and used in Synchro, where d_i is the delay per vehicle (s/veh) for movement i; g_i is the effective green time per cycle (s) for movement i; C is the optimal cycle length (s) for the intersection; s_i is the saturation flow (veh/s) for movement i and depends on the lanes' capacity; and $X_i = \frac{C}{g_i} \times \frac{q_i}{s_i}$ with q_i (veh/s) the arrival-rate indicates the movement's degree of saturation. The total delay D at the intersection is the sum $D = \sum_{i=1}^{12} d_i$. Now, let D_{α_j} be the delay assuming the arrival-rates q_i for each movement are equal to their predicted α_j-quantile $\tilde{y}_i^{(\alpha_j)}$, $i = 1,\ 2, \ldots,\ 12$. The aim is to compute

$$\{g_i^{opt}\}_{i=1}^{12},\ C^{opt} = \underset{\{g_i\}_{i=1}^{12},\ C}{\operatorname{argmin}} \quad \sum_{j=1}^{q} D_{\alpha_j} \quad , \tag{20}$$

where $\{\alpha_j\}_{j=1}^{q} = \{0.10,\ 0.30,\ 0.50, 0.70,\ 0.90\}$. This is a convex optimization problem and can be readily solved [15]. Once $\{g_i^{opt}\}_{i=1}^{12}$ and C^{opt} have been found, the realized total delay D caused by this combination of green splits and cycle length are calculated by setting $g_i = g_i^{opt}$ and $C = C^{opt}$ in (19) and letting the q_i's be equal to the actual flows.

For comparison, green splits are computed using empirical historical quantiles calculated over the entire data set. Table II records the delay engendered when adjusting the control policy according to the predicted quantiles compared to the data set quantiles. At the beginning of each hour, new green splits and cycle length are implemented following the procedure described above, using total flow quantiles over the next hour. More specifically, the table shows the estimated total delay for February 24th 2015 and July 2nd 2015 between 10:00 and 24:00, as well as the mean total delay for this time range across the entire data set. Additionally, a theoretical lower bound on the delay time is computed by supposing the actual flow is known in advance and optimizing for the actual flow. As expected, the predicted quantiles lead to much lower delays in comparison to the historical quantiles. On February 24, 2015 and July 2, 2015, total delay has been reduced by 5.9 hours and 4.3 hours respectively. Over the whole data set, total delay is decreased by 4.6 hours per day on average when implementing the control policy according to the predicted quantiles.

5 CONCLUSIONS

This project developed a powerful method for estimating quantiles of future traffic flow at an intersection using diverse real-time measurements. Furthermore, the efficiency of the regression algorithm was demonstrated through a case study conducted using data on a test site in South Carolina, although the techniques are applicable to any intersection capable of measuring traffic volume in real time. California possesses many such intersections, including many intersections in LA County. The predictions accurately described the observed traffic flows for several volume scenarios, using only computationally non-intensive operations. The case studies results demonstrated an average delay reduction of 4.6 hours per day at the intersection switching from a historical quantile-based control policy to a prediction-based policy. Through this algorithm, it is also possible to accomplish better green split management and reduce traffic delays while making no additional adjustments to the existing infrastructure encountered on the roads. The relative ease of implementation of the apparatus exposed in this paper makes quantile regression a versatile tool, handily applicable to a wide array of forecast-dependent fields. Future works could explore other types of regression, including parametric and nonparametric probability density fitting. Another interesting extension would be to examine the potential of quantile regression in the case of networked intersections. In this setting, can PLS capture the existing spatial correlations between contingent movements?

Moreover, since quantile predictions reflect historical, day-to-day variation in traffic flow, it may be possible detect anomalous deviations from usual traffic conditions due to car accidents or lane closures. Contributions could also be made to the theory of stochastic control in the framework of model-based traffic control design.

6 ACKNOWLEDGMENTS

Sensys Networks, Inc. and Beaufort County, South Carolina provided access to traffic flow data.

References

[1] A. A. Kurzhanskiy and P. Varaiya, "Traffic management: An outlook," *Economics of Transportation*, vol. 4, no. 3, pp. 135–146, 2015.

[2] V. Jain, A. Sharma, and L. Subramanian, "Road traffic congestion in the developing world," in *Proceedings of the 2nd ACM Symposium on Computing for Development*. ACM, 2012, p. 11.

[3] H. Chen, B. Jia, and S. Lau, "Sustainable urban form for chinese compact cities: Challenges of a rapid urbanized economy," *Habitat international*, vol. 32, no. 1, pp. 28–40, 2008.

[4] L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive kalman filtering for multi-step ahead traffic flow prediction," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 4724–4729.

[5] C.-J. Wu, T. Schreiter, and R. Horowitz, "Multiple-clustering armax-based predictor and its application to freeway traffic flow prediction," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 4397–4403.

[6] S. Coogan, C. Flores, and P. Varaiya, "Traffic predictive control from low-rank structure," *Transportation Research Part B: Methodological*, vol. 97, pp. 1–22, 2017.

[7] R. Juban, H. Ohlsson, M. Maasoumy, L. Poirier, and J. Z. Kolter, "A multiple quantile regression approach to the wind, solar, and price tracks of gefcom2014," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1094–1102, 2016.

[8] A. Haoui, R. Kavaler, and P. Varaiya, "Wireless magnetic sensors for traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 3, pp. 294–306, 2008.

[9] R. Koenker and G. Bassett Jr, "Regression quantiles," *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.

[10] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, latent structure and feature selection*. Springer, 2006, pp. 34–51.

[11] A. G. Bors, "Introduction of the radial basis function (rbf) networks," in *Online symposium for electronics engineers*, vol. 1, no. 1, 2001, pp. 1–7.

[12] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[13] R. Koenker, “Regression quantiles,” *Cambridge university press*, no. 38, pp. 5–15, 2005.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[15] R. E. Allsop, “Delay-minimizing settings for fixed-time traffic signals at a single road junction,” *IMA Journal of Applied Mathematics*, vol. 8, no. 2, pp. 164–185, 1971.

[16] Trafficware, “Synchro studio,” http://www.trafficware.com/synchro-studio.html, 2015.

[17] H. C. Manual, “Highway capacity manual,” *Washington, DC*, 2000.

Algorithm 1: Quantile Parameters Regression

```
Input     : Set of training input traffic flows $\{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^m$, collected from
            0:00 to $T_S$ on day $i$, set of quantiles $\{\alpha_1, \alpha_2, \ldots, \alpha_q\}$ to be computed
            with $\alpha_i \in [0,1]$, set of training output traffic flows $y \in \mathbb{R}^n$ with $y_i$
            collected at time $T_P$ on day $i$ ($T_P > T_S$), number of PLS components
            $m'$, number of k-means centers $k$, regularization parameter $\lambda \in \mathbb{R}$,
            ADMM step size $\delta \in \mathbb{R}$, number of iteration $T \in \mathbb{N}$
Output    : Set of quantile estimators $\Theta \in \mathbb{R}^{k \times q}$, k-means centers and
            bandwidths $\{(\mu_i, \sigma_i)\}_{i=1}^k$, matrix of PLS components $P$, mean-flow vector
            $\bar{x}$
Initialize: $A^1 = 0_{n,q},\ \ Z^1 = 0_{n,q},\ \ \Theta^1 = 0_{k,q}$
            $z_i^1$ denotes the $i$th column of $Z^1$

Compute mean-centered, aggregated data matrices
$\hat{X}$ and $\hat{Y}$, and mean flow vector $\bar{x}$ from (2)-(4);
Compute score matrix $S$ and component matrix $P$
using (5) to (13);
$\{(\mu_i, \sigma_i)\}_{i=1}^k =$ k-means++$(S,\ k)$;
Compute the stacked matrix of feature vectors $\Phi(S, k)$
according to (15) and (16);

Find the Cholesky decomposition $UU^T$ of $(\Phi^T\Phi + \frac{\lambda}{\delta}I)$;
for $j = 1,\ 2, \ldots,\ T$ do
    $\Theta^{j+1} = U^{-T}U^{-1}\Phi^T(y\mathbb{1}_q + Z^j - A^j)$;
    $\tilde{Z} = (\Phi\Theta^{j+1} - y\mathbb{1}_q + A^j)$;
    for each column $\tilde{z}_l$ of $\tilde{Z}$, $l = 1,\ 2, \ldots,\ q$ do
        $z_l^{j+1} = \max\{0,\ \tilde{z}_l - \frac{1}{\delta}\alpha_l\} + \min\{0,\ \tilde{z}_l - \frac{1}{\delta}(\alpha_l - 1)\}$;
        (this is a component-wise operation)
    end
    $A^{j+1} = A^j + \Phi\Theta^{j+1} - y\mathbb{1}_q - Z^{j+1}$ ;
end
return $\Theta,\ \{(\mu_i, \sigma_i)\}_{i=1}^k,\ P,\ \bar{x}$
```

Algorithm 2: Quantile Predictions and Score Computation

```
Input  : Test input flow $\hat{x} \in \mathbb{R}^m$ collected from 0:00 to $T_S$, test output flow
         $\hat{y} \in \mathbb{R}$ measured at time $T_P$ ($T_P > T_S$), set of quantile estimators
         $\Theta = [\theta_1 \ \ldots \ \theta_q] \in \mathbb{R}^{k \times q}$, set of RBF centers and bandwidths
         $\{(\mu_i, \sigma_i)\}_{i=1}^{k}$, matrix of PLS components P, mean-flow vector $\bar{x}$
Output: Set of predicted quantiles $\{\tilde{y}^{(\alpha_1)}, \tilde{y}^{(\alpha_2)}, \ldots, \tilde{y}^{(\alpha_q)}\}$, prediction score $\epsilon$

$\hat{S} = H(\hat{x}) = ((\hat{x} - \bar{x})^T (P^T)^\dagger)^T$;
$T(\hat{x}) = \phi(\hat{S})$ according to (10);
for $i = 1, 2, \ldots, q$ do
    $\tilde{y}^{(\alpha_i)} = \theta_i^T T(\hat{x})$;
end

Sort $\{\tilde{y}^{(\alpha_1)},\ \tilde{y}^{(\alpha_2)}, \ldots,\ \tilde{y}^{(\alpha_q)}\}$ in ascending order;
$\epsilon = \sum_{i=1}^{q} \rho_{\alpha_i}(\tilde{y}^{(\alpha_i)} - \hat{y})$;

return $\{\tilde{y}^{(\alpha_i)}\}_{i=1}^{q},\ \epsilon$
```

	Tilted loss score ($\times 10^2$)		
	Historical	Predicted	Improvement
February 24, 2015	253	51	202
July 2, 2015	230	45	185
January 1, 2015	444	35	409
Data Set Mean	60	38	22

Table 1: Comparison of tilted loss function total scores obtained from predicted quantiles and historical quantiles on three different test days. Hourly flows from 10:00 onward were chosen as targets for prediction. The total scores are computed by summing the individual daily scores for each movement. The predictions' performance surpasses that of the historical quantiles, with an average daily titled loss score of 38×10^2 vs. 60×10^2.

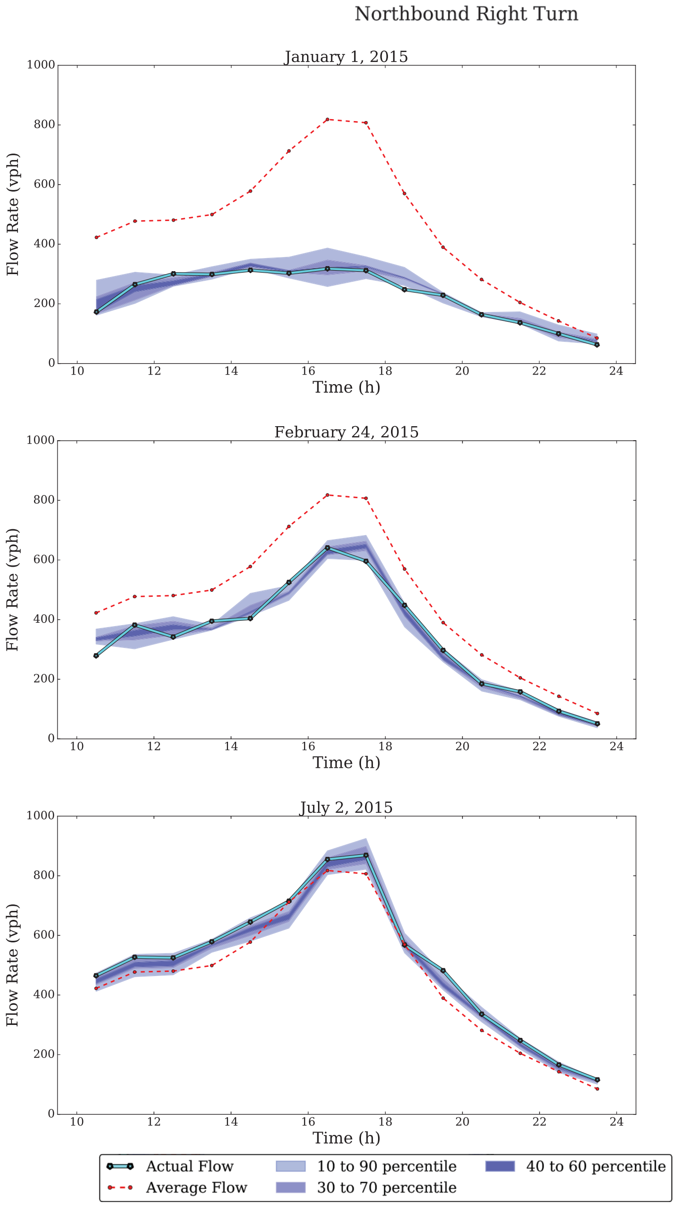

Figure 3: Example of predictions for the NB-RT movements on three days with different traffic profiles. Lightest blue indicates the predicted 10 to 90 percentile range, with darker tones corresponding to the 30 to 70 and 40 to 60 ranges. The algorithm successfully predicts well below average traffic on the holiday of January 1, 2015. Furthermore, it correctly predicts below average traffic due to winter weather on February 24, 2015, as well as above average traffic on July 2, 2015.

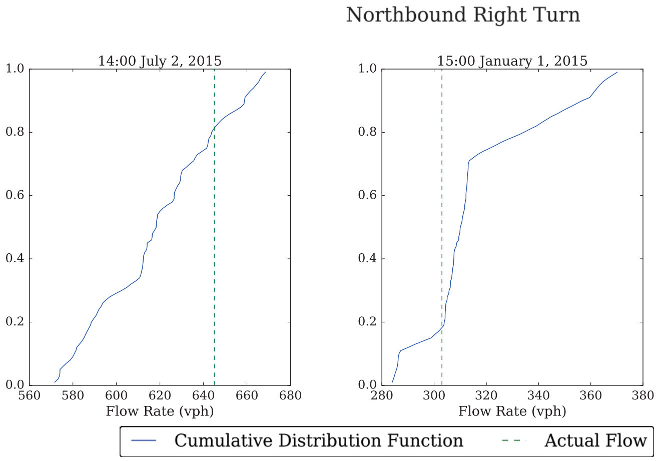

Figure 4: Plots of predicted cumulative distribution functions for two separate times experiencing unalike traffic volumes. Steeper slopes indicate higher expected probabilities of occurrence. Observed flows were accurately predicted by the algorithm for both cases.

	Feb. 24, 2015	Jul. 2, 2015	Data Set Mean
Delay using Historical Quantiles (h)	99.5	275.7	181.2
Delay using Predicted Quantiles (h)	93.6	271.4	176.6
Delay lower bound (h)	91.8	269.3	173.5
Predicted vs. Historical improvement (h)	5.9	4.3	4.6

Table 2: Illustrative total delays estimated for two test days between 10:00 and 24:00. Additionally, the average total delay across the data set is displayed. The delays are computed using both predicted and empirical historical quantiles; a lower bound on the total nominal delays was also calculated. Adjusting the green cycles according to the predictions improves total delay by 4.6 hours per day.