# REAL-TIME ESTIMATION OF INTERSECTION TURNING PROPORTIONS FROM EXIT COUNTS


Dongyan Su (corresponding author)
University of California, Bekeley
2167c Etcheverry Hall
University of California
Berkeley, CA 94720-1740
Tel: 1-510-928-8771
dongyan@berkeley.edu

Roberto Horowitz
University of California, Bekeley
5138 Etcheverry Hall
University of California
Berkeley, CA 94720-1740
Tel: 1-510-642-4675
horowitz@me.berkeley.edu

Xiao-Yun Lu
PATH, ITS, Universion of California, Berkeley
RFS, Bldg 452, 1357 S. 46th Street
Richmond, CA 94804-4648
Tel: 1-510-665-3644
xylu@path.berkeley.edu



Submission Data: July 31, 2013
Word Count: 5833 (4083 words + 4 tables + 3 figures)

## Abstract

Estimation of intersection origin-destination (OD) matrices is a frequently-encountered problem in transportation management operations. In particular, advanced traffic responsive arterial signalization management schemes rely on the accurate real-time estimation of vehicle queue lengths and turning ratios. This paper proposes a new method for estimating OD matrices of a signalized intersection in real time. The proposed method only requires detectors at exit legs, saving half detectors from a complete configuration assumed by previous research. Signal phase timing information is used to determine the portion of the volume measurement that corresponds to a specific phase. The estimation of turning proportions is formulated as a constrained least squares problem and solved using recursive algorithm with forgetting factor and covariance resetting. Simulation shows that the proposed method can obtain accurate results and response to parameter changes fast, while requiring significant less computational effort than a comparable solution using a constrained optimization method.

# I - INTRODUCTION

In advanced intersection signal operations, one frequent requirement is the real-time knowledge of turning volumes, or turning proportions (1). This information is usually described by the so-call origin-destination (OD) matrices. In the past decades, great attention has been received in estimating OD matrices from traffic counts collected by sensors, and a lot of methods have been proposed (2).

In most of the previous research, a complete configuration of detectors was assumed to be available. A complete configuration means detectors are installed at all entrance legs and exit legs for measuring traffic volumes. In such a configuration, the problem of estimating turning proportions could be formulated as a linear system with linear equality and inequality constraints. This system is underdetermined, and least squares method could be used to solve it. To solve a constrained least squares problem, people can use a variety of constrained quadratic optimization algorithms, or routine in (3). However, a direct solution of constrained least squares problems through optimization could be computationally expensive, thus inappropriate for real-time practice. With this concern, different recursive approaches have been proposed. Early Examples can be found in (4, 5). Those recursive methods usually included two major steps. In the first step, unconstrained least squares estimator is used. In the second step, parameter estimates are constrained using some correction methods, such as normalization and truncation. Because of the operation of normalization or truncation, there is no guarantee that the resulting estimates are optimal. Motivated by this concern, (6) developed a simple algorithm for the least squares problem with inequality boundary constraints. Unlike the previous methods, constraints were enforced by an iteration process in this algorithm. The convergence of this algorithm was proven, and it was stated that the imposition of the inequality constraints in the update iteration could improve the accuracy. Based on this algorithm, (7) proposed a recursive method which further incorporated both equality and boundary inequality constraints in the least squares update. This method could achieve good estimation in much smaller computational effort, compared to an optimization approach.

Even though recursive approaches have been shown to be computationally efficient and well-performed, there are still two concerns about these methods. The first concern is that a complete configuration of detectors may not be available. In a complete configuration, a typical four-way intersection requires at least eight detectors, whose price might be expensive. In order to get an economic configuration, a “Cordon” configuration, which groups several intersections and deletes one or more intermediate detectors, can be used. (8) firstly introduced a condition to examine the feasibility of estimating from an incomplete set of traffic counts. As long as this condition is met, it is able to obtain estimates from the incomplete information. But it was also mentioned in (8) that an increase in variability could happened since less traffic counts were accessible. Subsequently, (9) suggested that the condition number of the Jacobian matrix, which is derived from predicted output counts with respect to the parameters, could be used to evaluate whether the incomplete information was sufficient to provide “good” estimates. In its numerical test, it was shown that the condition number enlarged dramatically as detector saving grew, which might lead to larger potential of poor estimation. Some real-time estimation strategies addressing incomplete detector configuration can be found in (10, 11, 12).

The second concern of some previously proposed methods is that they might fail to detect changes in parameters. Most of the recursive methods were designed to estimate dynamic turning proportions in real time. To achieve this, time-series of data were used. However, most of those methods, including some mentioned above, were only evaluated with static turning proportions in the paper they presented. Performance of these methods was not shown for the case that turning proportions changed during simulation. Some research has shown that the recursive least squares estimator parameter adaptation gain decreases as time progresses. Thus, an ordinary least squares estimator may fail to follow parameter changes in a dynamic scenario. The technique of forgetting factor or covariance resetting has been suggested by researchers for tracking time-varying parameters. Description of these techniques can be found in (13, 14).

This paper presents a new method for estimating turning proportions in real time. Compared with previous research, the algorithm proposed in this paper only requires detectors to be installed at exit legs. Hence, it saves half detectors from a complete detector configuration. However, real-time signal is used to relate the portion of traffic volumes to its corresponding phase. The estimation problem is formulated as a constrained least squares problem, and then solved recursively. Therefore, it is called Recursive Constrained Least Squares (RCLS). An extended version of RCLS is presented for the case of time-varying turning proportions. It applies forgetting factor and covariance resetting. This extended version is called Recursive Constrained Least Squares with Forgetting Factor and Covariance Resetting (RCLSFR).

The reminder of the paper is organized as follows: Section II describes the mathematical formulation of the estimation problem. Section III explains the steps used to solve the problem. Section IV discusses how to introduce a forgetting factor and covariance resetting in the parameter adaptation algorithm. Section V evaluates the algorithm using simulation. Conclusion is presented in Section VI.

## II - PROBLEM FORMULATION

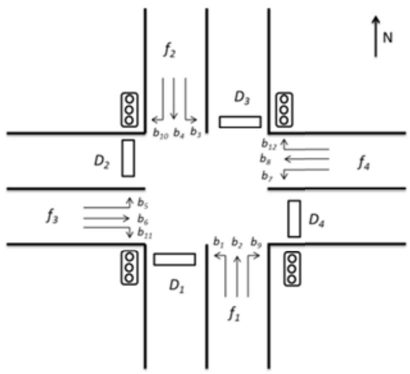


**FIGURE 1 Intersection with departure detectors.**

Fig. 1 shows a four-way signalized intersection with detectors presented at exit legs. The northbound and southbound arrival volumes are denoted as $f_1$ and $f_2$. $b_1$ and $b_2$ are used to represent the northbound left and through percentages, and $b_3$ and $b_4$ to represent the southbound left and through percentages. As a consequence, the northbound and southbound right-turn percentages, denoted as $b_9$ and $b_{10}$, equal to 1-$b_1$-$b_2$ and 1-$b_3$-$b_4$, respectively. Similar notations are used for eastbound and westbound. Assume that there is no right-turn on red, and the measurements are noise-free. During the period that northbound traffic and southbound traffic get green, the numbers of vehicles passing each detector are denoted as $D_1$ to $D_4$. From flow conservation, it is straightforward to obtain

$$f_2 b_4 = D_1 \tag{1}$$

$$f_1 b_1 + f_2 \left(1 - b_3 - b_4\right) = D_2 \tag{2}$$

$$f_1 b_2 = D_3 \tag{3}$$

$$f_1 (1 - b_1 - b_2) + f_2 b_3 = D_4 \tag{4}$$

In the period that eastbound traffic and westbound traffic get green, another set of traffic counts should be collected. One can easily obtain the same equations, only that the subscripts will change. Therefore, the ideas discussed below apply to both north-south-bound traffic and east-west-bound traffic, with the only difference subscripts. Hence, in the discussion that follows, the north-south-bound traffic will be taken as a representative and east-west-bound traffic will not be discuss further.

Notice that no assumption is made regarding the lane configuration of intersections, or the type of signal control. Having a designated lane for a particular movement or not, or being a pre-time control or actuated control, will not change equation (1) - (4). However, it is necessary for north-south-bound and east-west-bound traffic not to receive green at the same time, which is frequently the case with most signalized intersections.

Using (1) and (3) to substitute $f_1$ and $f_2$ in (2) and (4), one obtains

$$D_3 \frac{b_1}{b_2} + D_1 \left( \frac{1}{b_4} - \frac{b_3}{b_4} - 1 \right) = D_2 \quad (5)$$

$$D_3 \left( \frac{1}{b_2} - \frac{b_1}{b_2} - 1 \right) + D_1 \frac{b_3}{b_4} = D_4 \quad (6)$$

Because all the $b_i$'s must fall in the interval of [0, 1], one have the constraints

$$\frac{1}{b_2}, \frac{1}{b_4} \geq 1 \quad (7)$$

$$\frac{1}{b_2} \geq \frac{b_1}{b_2} \geq 0 \quad (8)$$

$$\frac{1}{b_4} \geq \frac{b_3}{b_4} \geq 0 \quad (9)$$

Also, because $b_1 + b_2 \leq 1$ and $b_3 + b_4 \leq 1$, there are another two constraints

$$\frac{b_1}{b_2} - \frac{1}{b_2} \leq -1 \quad (10)$$

$$\frac{b_3}{b_4} - \frac{1}{b_4} \leq -1 \quad (11)$$

Constraints (7) - (11) can be rewritten as

$$\frac{1}{b_2}, \frac{1}{b_4} \geq 1 \quad (12)$$

$$\frac{b_1}{b_2}, \frac{b_3}{b_4} \geq 0 \quad (13)$$

$$\frac{1}{b_2} - \frac{b_1}{b_2} \geq 1 \quad (14)$$

$$\frac{1}{b_4} - \frac{b_3}{b_4} \geq 1 \quad (15)$$

Equation (5), (6) and constraints (12) - (15) define a problem for $b_1$ to $b_4$. There are four unknowns, two equations, and six inequalities. This is an underdetermined problem. It can be solved in the least squares sense.

Since (5), (6), (12) - (15) are not in a good form to apply least squares estimator, they are reorganized into a matrix form (16) and (17).

$$\begin{bmatrix} 0 & D_3 & D_1 & -D_1 \\ D_3 & -D_3 & 0 & -D_1 \end{bmatrix} \begin{bmatrix} \frac{1}{b_2} - 1 \\ \frac{b_1}{b_2} \\ \frac{1}{b_4} - 1 \\ \frac{b_3}{b_4} \end{bmatrix} = \begin{bmatrix} D_2 \\ D_4 \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} \frac{1}{b_2} - 1 \\ \frac{b_1}{b_2} \\ \frac{1}{b_4} - 1 \\ \frac{b_3}{b_4} \end{bmatrix} \geq 0 \tag{17}$$

Let $\beta = \begin{bmatrix} \frac{1}{b_2} - 1, & \frac{b_1}{b_2}, & \frac{1}{b_4} - 1, & \frac{b_3}{b_4} \end{bmatrix}^T$, $X = \begin{bmatrix} 0 & D_3 & D_1 & -D_1 \\ D_3 & -D_3 & 0 & -D_1 \end{bmatrix}$ and $Y = \begin{bmatrix} D_2 \\ D_4 \end{bmatrix}$, the problem (16) and (17) can be represented by

$$Y = X\beta \tag{18}$$

Subject to

$$\beta \geq 0 \tag{19}$$

In practice, time-series of data are used, and detector measurements might not be accurate due to noise or vehicle traverse delay. A more exact expression of equation (18) should be

$$Y(k) = X(k)\beta + v(k) \tag{20}$$

Here $k$ represents time index and $v$ represents measurement error. To solve for $\beta$ in the least squares sense, one minimizes the object function (21).

$$\min_{\hat{\beta}} \sum_{k=1}^{N} \left( Y(k) - X(k)\hat{\beta} \right)^T \left( Y(k) - X(k)\hat{\beta} \right) \tag{21}$$

The hat on the variable denotes that it is an estimated value. (19) - (21) define a least squares problem with inequality boundary constraints. It can be solved by an algorithm proposed by Bell (6) and Li (7). Notice that constraints (10) and (11) are excluded in (17), because it will bring in linear constraints other than boundary constraints, and consequently makes computation difficult. The problem will be solved in the following procedure. In the first step, an ordinary least squares estimator is used to solve (20). In the second step, Bell's correction will be applied if constraint (19) is not met. In the third step, $\hat{b}_i$'s are solved from $\hat{\beta}$. Within this step, normalization and truncation will be conducted if constraints (10) and (11) are not satisfied. Details of the solving procedure can be found in next section.

## III - STEPS TO SOLVE THE LEAST SQUARES PROBLEM

**Step 0**: Initialize $\hat{\beta}_0$ and $P_0$. $\hat{\beta}_0$ is a 4×1 nonnegative vector. It can be initialized by prior knowledge, or random values. $P_0$ is a 4×4 positive matrix.
**Step 1**: Update by an ordinary least squares estimator. After collecting new traffic counts, construct $X_k$ and $Y_k$, and obtain the estimates $\hat{\beta}_k$ by

$$\hat{\beta}_k = \hat{\beta}_{k-1} + K_k(Y_k - X_k\hat{\beta}_{k-1}) \tag{22}$$

Where

$$K_k = P_{k-1}X_k^T S_k^{-1} \tag{23}$$

$$S_k = X_k P_{k-1} X_k^T + I \tag{24}$$

$$P_k = (I - K_k X_k)P_{k-1} \tag{25}$$

Here $I$ is the identity matrix.
**Step 2**: If there is any negative value in $\hat{\beta}_k$, apply Bell's correction to guarantee the nonnegative constraint is met. Otherwise, skip this step.

2.1 Let $\tilde{\beta}_k = \hat{\beta}_k$ and $\mu = 0$.

2.2 Compute $D = diag\{1/p_{ii}\}$, where $P_k = [p_{ij}]_{4\times 4}$.

2.3 Compute $\mu = TF(\mu - D\tilde{\beta}_k)$. Here $TF(\bullet)$ is a truncation function which drives all negative values to zeros.

2.4 $\tilde{\beta}_k = \hat{\beta}_k + P_k\mu$.

Repeat 2.3 and 2.4 until $\tilde{\beta}_k$ converges. If step 2 is executed, $\hat{\beta}_k$ will be replaced by $\tilde{\beta}_k$.
**Step 3**: Solve for $\hat{b}_i$'s from $\hat{\beta}_k$. If there is any $\hat{b}_i$ being negative or greater than one, truncate it to zero or one. If $\hat{b}_1 + \hat{b}_2 > 1$, normalize $\hat{b}_1$ and $\hat{b}_2$ so that $\hat{b}_1 + \hat{b}_2 = 1$. Apply normalization if $\hat{b}_3 + \hat{b}_4 > 1$. Compute the turning proportions for right-turns $\hat{b}_9$ and $\hat{b}_{10}$ by $1 - \hat{b}_1 - \hat{b}_2$ and $1 - \hat{b}_3 - \hat{b}_4$, respectively.

The above steps, Step 0 to Step 3, are the solving procedure in RCLS. Step 1 to step 3 are repeated if new measurements are available.

## VI - INCORPORATE FORGETTING FACTOR AND COVARIANCE RESETTING

Research has shown that the gain ($K_k$ in equation (22)) in an ordinary least squares estimator would become smaller and smaller as the number of steps grows. So an ordinary least squares estimator might fail to trace time-varying variables. If the turning proportions change during a studied period, the extended version of RCLS, RCLSFR, should be used. In RCLSFR, forgetting factor and covariance resetting are applied in the covariance update. In this case, equation (26) is used instead of (25) in Step 1.

$$P_k = \frac{1}{\lambda}\left(I - K_k X_k\right)P_{k-1} + \varepsilon I - \delta P_{k-1}^2 \tag{26}$$

In (26), $\lambda$ is the forgetting factor, and it usually falls in [0.9, 1]. $\varepsilon$ and $\delta$ are small values to adjust covariance resetting. They are in [0, 0.1] in usual practice.

Notice that, in the case that equation (26) is used in Step 1, the matrix $P_k$ in Step 2 should still be updated through equation (23) - (25). As a result, two different sets of matrices $K_k$, $S_k$, and $P_k$ have to be maintained during the solving procedure.

## V - SIMULATION AND RESULTS

In this section, the accuracy and computational effort of the proposed method will be evaluated by simulation. The estimated turning proportions from the proposed approach, RCLS or RCLSFR, and those from an optimization approach will be compared. The optimization approach chosen is the Constrained Least Squares by Active-Set algorithm (CLSAS) in Matlab 2013a. Two scenarios, one with static turning proportions and the other with time-varying proportions, will be simulated.

**Scenario I**

In this scenario, static turning proportions will be used to compare RCLS and CLSAS. Consider a four-way intersection with turning proportions shown in Table 1 (source from (11)). Arrival volumes are random variables generated from Poisson distribution with mean 100. Turning volumes are multinomial random numbers generated from arrival volumes and turning proportions define in Table 1. Exit volume at each exit leg would be the sum of all turning volumes entering that leg, denoted as $q_i$. The measurement error is modeled as Gaussian distribution with standard deviation equal to 10% of the exit volume. Hence, the traffic count obtained at a detector $D_i$ is computed as

$$D_i(k) = q_i(k) + v_i(k) \tag{27}$$

Where $v_i(k) = N(0, 0.1q_i)$. $D_i$ will be rounded if it is not a nonnegative integer.

**TABLE 1 Turning Proportions**

| | Left | Straight | Right |
|---|---|---|---|
| Northbound | 0.23 | 0.414 | 0.356 |
| Southbound | 0.29 | 0.352 | 0.29 |
| Eastbound | 0.149 | 0.8 | 0.051 |
| Westbound | 0.083 | 0.843 | 0.074 |

Traffic counts for 10 intervals are generated in simulation. A constrained least squares problem defined by equations (5), (6) and constraints (12) - (15) is solved in every interval by optimization approach CLSAS. It takes as input all the traffic counts generated from beginning to current interval. The proposed method RCLS is also executed every interval. Its inputs are the latest traffic counts and the estimated variables from previous interval. The initial estimate $\hat{\beta}_0$ in RCLS is obtained by random positive values.

To evaluate the accuracy, the error between estimated turning proportions and the real ones is computed at the end of each simulation run. RMSD (root-mean-square deviation) is chosen as the indicator of error. The definition of RMSD is shown in equation (28). The time for executing an algorithm on 10 intervals of data is collected in order to compare the computational effort. Furthermore, the frequencies of applying corrections (step 2 and step 3) are used to indicate the effort RCLS pays to satisfy the constraints, after its update from least squares estimator. Variables of step requiring iteration, number of iterations, and step requiring truncation are computed in simulation. Step requiring iteration is the times when step 2 is applied in a simulation run. Number of iterations is the total count of iteration executed in a run, that is, how many times step 2.3 is needed to get rid of the nonnegative values. Step requiring truncation computes the times when any truncation or normalization is applied in step 3.

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}(b_{i,estimated} - b_{i,real})^2}{N}} \tag{28}$$

Table 2 shows the performance comparison between CLSAS and RCLS, and Figure 2 plots the RMSD curves in experiment no. 1 to no. 4. It could be seen from Table 2 that the values of RMSD in CLSAS and in RCLS are very close. Six out of ten experiments have RMSD in RCLS same as in CLSAS, and the remaining have the RMDS difference only at the last digit. It is obvious from Figure 2 that, after the first few steps, the RMSD curve of RCLS almost overlaps with that of CLSAS in each experiment. The execution time of RCLS is only about 3% of that of CLSAS. This means that RCLS can get as good estimation as CLSAS in a much smaller computational effort. It can also be found that, on average, the iteration in step 2 is only executed 0.9 times out of 20 steps (two directions, and 10 intervals of data for each direction). And only 3.3 iterations are applied in one experiment. Truncation or normalization is conducted 2.4 times out of 20 steps. All these values indicate that the ordinary least squares estimator in RCLS gets an estimate meeting the constraints most of time, and the iteration procedure in its algorithm will not cost great computational effort, and therefore not influence real-time efficiency.

**TABLE 2 Performance Comparison of CLSAS and RCLS**

| | CLSAS | | RCLS | | | | |
|---|---|---|---|---|---|---|---|
| Experiment NO. | RMSD | Execution time | RMSD | Execution time | Step requiring iteration | Number of iteration | Step requiring truncation |
| 1 | 0.0083 | 0.0919 | 0.0083 | 0.0032 | 1 | 2 | 3 |
| 2 | 0.0359 | 0.0938 | 0.0358 | 0.0032 | 1 | 2 | 1 |
| 3 | 0.0247 | 0.1438 | 0.0247 | 0.0022 | 0 | 0 | 3 |
| 4 | 0.0340 | 0.0989 | 0.0338 | 0.0027 | 2 | 4 | 3 |
| 5 | 0.0090 | 0.0962 | 0.0091 | 0.0031 | 2 | 19 | 5 |
| 6 | 0.0142 | 0.0982 | 0.0142 | 0.0025 | 1 | 2 | 1 |
| 7 | 0.0235 | 0.0893 | 0.0235 | 0.0027 | 0 | 0 | 1 |
| 8 | 0.0132 | 0.0967 | 0.0134 | 0.0022 | 1 | 2 | 2 |
| 9 | 0.0155 | 0.0904 | 0.0154 | 0.0022 | 1 | 2 | 2 |
| 10 | 0.0254 | 0.0963 | 0.0254 | 0.0031 | 0 | 0 | 3 |
| Mean | 0.0203 | 0.0996 | 0.0203 | 0.0027 | 0.9 | 3.3 | 2.4 |

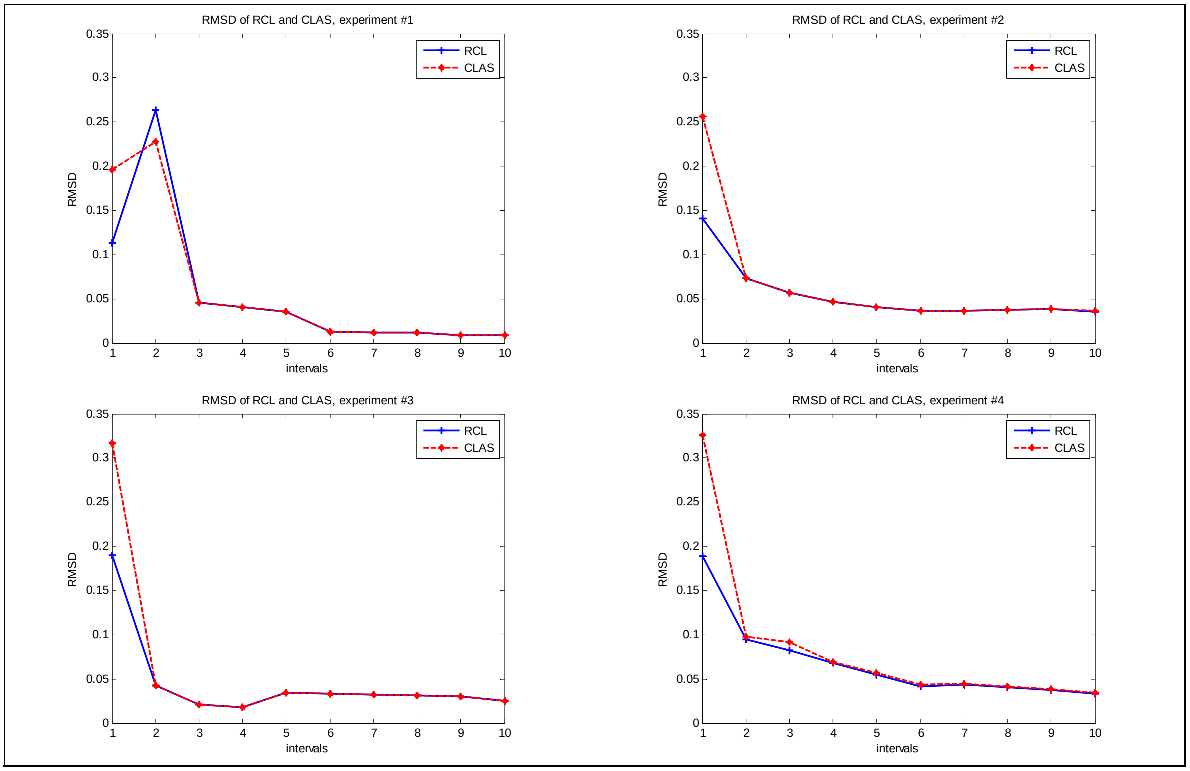


**FIGURE 2 Example result of experiment #1 to #4.**

## Scenario II

In this scenario, RCLS, CRLSFR, and CLSAS will be compared in estimating time-varying turning proportions. The following simulation is conducted for the comparison. Consider the same intersection used above. The initial turning proportions are defined in Table 1 and kept for 20 intervals. These proportions are followed by the turning proportions defined in Table 3, which are kept for another 20 intervals. The process mentioned above is used to generate arrival volumes, exit volumes, and detector counts. RCLS and CLSAS are tested in the same way as above. However, adjustment has to be made to test CLSAS in this scenario. This is because, if all the traffic counts generated from beginning to current interval are input to CLSAS optimization, as what we do in the previous scenario, CLSAS will compute the aggregated turning proportions in the second period, rather than the new ratios. Only using the latest traffic counts is neither a good idea, because the estimates will be significantly affected by the randomness or noise. With such concerns, it has to be decided how many pieces of data should be fed into the optimization. From some empirical observations, it is found that using eight intervals of data gives best estimate in this test. As a result, CLSAS takes the latest 8 pieces of data in the experiments of this part. The parameters $\lambda$, $\varepsilon$ and $\delta$ in RCLSFR also need to be determined. $\lambda$=0.995, $\varepsilon$=0.0005 and $\delta$=0.0005 are chosen because they give best estimate in tuning test.

**TABLE 3 Turning Proportions in Second Period**

| | Left | Straight | Right |
|---|---|---|---|
| Northbound | 0.32 | 0.544 | 0.136 |
| Southbound | 0.16 | 0.471 | 0.369 |
| Eastbound | 0.33 | 0.54 | 0.13 |
| Westbound | 0.2 | 0.5 | 0.3 |

**TABLE 4 Performance Comparison of CLSAS, RCLS and RCLSFR**

| | CLSAS | | RCLS | | | | | RCLSFR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Experiment No. | RMSD | Execution time | RMSD | Execution time | Step requiring iteration | Number of iteration | Step requiring truncation | RMSD | Execution time | Step requiring iteration | Number of iteration | Step requiring truncation |
| 1 | 0.0326 | 0.2991 | 0.1323 | 0.0054 | 0 | 0 | 0 | 0.0299 | 0.0092 | 0 | 0 | 3 |
| 2 | 0.0271 | 0.3793 | 0.1332 | 0.0087 | 2 | 13 | 3 | 0.0559 | 0.0142 | 5 | 17 | 6 |
| 3 | 0.0483 | 0.3051 | 0.1238 | 0.0056 | 2 | 4 | 2 | 0.0580 | 0.0110 | 1 | 2 | 3 |
| 4 | 0.0247 | 0.2895 | 0.1120 | 0.0073 | 1 | 2 | 2 | 0.0337 | 0.0111 | 2 | 4 | 6 |
| 5 | 0.0629 | 0.2984 | 0.1387 | 0.0078 | 0 | 0 | 2 | 0.0310 | 0.0097 | 2 | 4 | 4 |
| 6 | 0.0211 | 0.2918 | 0.1159 | 0.0105 | 0 | 0 | 1 | 0.0360 | 0.0088 | 0 | 0 | 1 |
| 7 | 0.0265 | 0.2903 | 0.1202 | 0.0053 | 0 | 0 | 1 | 0.0250 | 0.0143 | 0 | 0 | 3 |
| 8 | 0.0510 | 0.3025 | 0.1074 | 0.0068 | 1 | 7 | 1 | 0.0237 | 0.0109 | 1 | 7 | 2 |
| 9 | 0.0126 | 0.3018 | 0.0909 | 0.0072 | 1 | 2 | 1 | 0.0192 | 0.0109 | 0 | 0 | 2 |
| 10 | 0.0121 | 0.3247 | 0.0915 | 0.0060 | 0 | 0 | 1 | 0.0327 | 0.0122 | 0 | 0 | 2 |
| Mean | 0.0319 | 0.3083 | 0.1166 | 0.0071 | 0.7 | 2.8 | 1.4 | 0.0345 | 0.0112 | 1.1 | 3.4 | 3.2 |

Table 4 shows the performance of CLSAS, RCLS and RCLSFR run in this scenario, and Figure 3 plots the RMSD curves in experiment no. 1 to no. 4 in this scenario. It can be noticed that the RMSD in RCLS is about three times larger than that in CLSAS. From Figure 3, it is also observed that, even though RCLS gives a good estimation in the first period, its RMSD jumps to a very high value and drops very slowly after the turning proportions change. These values and the pattern of RMSD curve indicate that, RCLS can hardly detect the change in parameters. On the other hand, RCLSFR has a much smaller RMSD than RCLS in the second period, yet still larger than that in RCLS. It is shown by Figure 3 that, the curve of RCLSFR drops faster than that of CLSAS after change happened, but it fluctuates later and ends up in a larger error than CLSAS. The pattern of RMSD curves demonstrates that RCLSFR is able to detect parameter change fast, while having some fluctuation in the stationary period. The computational time of RCLSFR is about 3% of CLSAS, slightly larger than RCLS. Values for step requiring iteration, number of iteration and step requiring truncation are 1.1, 3.4, and 3.2 respectively. As indicated by these values, RCLSFR does not pay much effort to satisfy the constraints after its update from least squares estimator. Given its fast response to change and small computational effort, RCLSFR is still a competitive algorithm compare to CLSAS, even for its fluctuations in the stationary period.

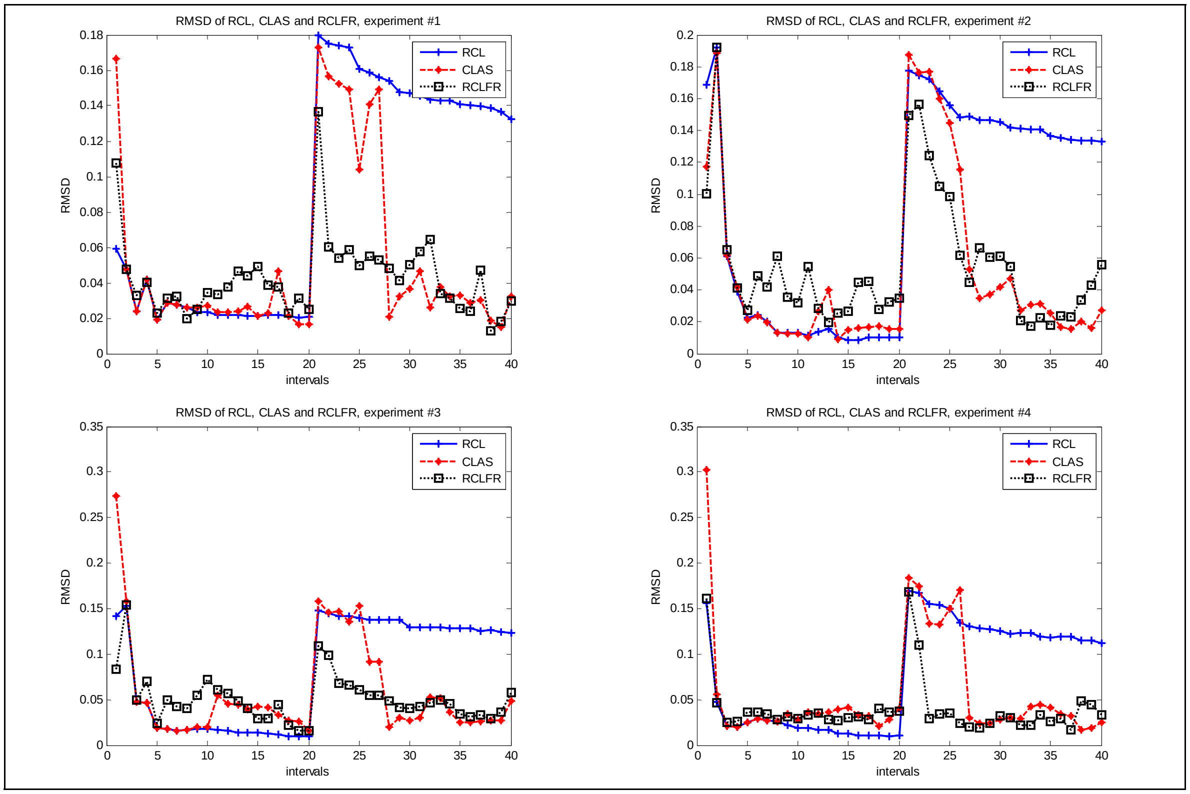


**FIGURE 3 Example result of experiment #1 to #4.**

## VI - CONCLUSION

A recursive method RCLS is proposed in this paper for estimating turning proportions of signalized intersections in real time. This method needs exit counts only, and it saves half detectors from a complete detector configuration. The estimation problem is in the form of constrained least squares, and a recursive algorithm is presented to solve it. Simulation shows that the proposed method can obtain as accurate estimation as the optimization approach CLSAS, while its computation is only 3% of CLSAS. The extended method RCLSFR, which incorporates forgetting factor and covariance resetting, is suggested for estimating time-varying turning proportions. It is able to response to change in parameters very quickly. Even though it has a bit larger error than CLSAS in the stationary period, it takes significantly less computational effort.

ACKNOWLEDGEMENTS

This work is partially supported by the California Department of Transportation (Caltrans) through the Connected Corridors California PATH Program and by the National Science Foundation (NSF) through grant CDI-0941326.

## REFERENCE

(1) Varaiya, Pravin. The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections. *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer New York, 2013, pp: 27-66.
(2) Willumsen, L. G. Estimation Of An OD Matrix From Traffic Counts–A Review. Institute for Transport Studies Working Paper 99, Leeds University (1978).
(3) Lawson, C. L., and R. J. Hanson. *Solving Least Squares Problems*. Vol. 161. Englewood Cliffs, NJ: Prentice-hall, 1974.
(4) Cremer, M., and H. Keller. A New Class of Dynamic Methods for the Identification of Origin-Destination Flows. *Transportation Research Part B: Methodological* 21.2 (1987), pp. 117-132.
(5) Nihan, N. L., and G. A. Davis. Recursive Estimation of Origin-Destination Matrices from Input/Output Counts. *Transportation Research Part B: Methodological* 21.2 (1987), pp. 149-163.
(6) Bell, M. GH. The Estimation of Origin-Destination Matrices by Constrained Generalised Least Squares. *Transportation Research Part B: Methodological*25.1 (1991), pp. 13-22.
(7) Li, B., and B. De Moor. Recursive Estimation Based on the Equality-Constrained Optimization for Intersection Origin–Destination Matrices.*Transportation Research Part B: Methodological* 33.3 (1999), pp. 203-214.
(8) Lan, C, and G. A. Davis. Real-Time Estimation of Turning Movement Proportions from Partial Counts on Urban Networks. *Transportation Research Part C: Emerging Technologies* 7.5 (1999), pp: 305-327.
(9) Lan, C. Sufficiency of Detector Information under Incomplete Configuration for Intersection OD Estimation. *Proceedings of Intelligent Transportation Systems, 2001*, pp: 398-403.
(10) Lan, C., and G. A. Davis. Real-Time Estimation of Turning Movement Proportions from Partial Counts on Urban Networks. *Transportation Research Part C: Emerging Technologies* 7.5 (1999), pp: 305-327.
(11) Li, B., and B. De Moor. Dynamic Identification of Origin–Destination Matrices in the Presence of Incomplete Observations. *Transportation Research Part B: Methodological* 36.1, 2002, pp: 37-57.
(12) Chang, G., and X. Tao. Estimation of Time-Dependent Turning Fractions at Signalized Intersections. In *Transportation Research Record: Journal of the Transportation Research Board No. 1644*, Transportation Research Board of the National Academies, Washington, D.C., 1998, pp:142-149.
(13) Gunnarsson, S.. Combining Tracking and Regularization in Recursive Least Squares Identification. *Decision and Control, 1996., Proceedings of the 35th IEEE*, vol. 3, pp. 2551-2552.
(14) Salgado, M. E., G. C. Goodwin, and R. H. Middleton. Modified Least Squares Algorithm Incorporating Exponential Resetting and Forgetting. *International Journal of Control* 47.2 (1988), pp: 477-491.