

# **A Robust Algorithm for Improved Dual-Loop Detection on Freeways**

Xiaoping Zhang  
PhD, PE  
DKS Associates  
Seattle, WA 98104  
Tel: (206) 587-0676  
Fax: (206) 587-0692  
Email: [xpz@dkssea.com](mailto:xpz@dkssea.com)

Nancy L. Nihan  
Professor and Director  
Transportation Northwest (TransNow)  
Department of Civil and Environmental Engineering  
University of Washington  
Seattle, WA 98195-2700  
Tel: (206) 543-9639  
Fax: (206) 543-5965  
Email: [nihan@u.washington.edu](mailto:nihan@u.washington.edu)

Yinhai Wang  
Assistant Professor  
Department of Civil and Environmental Engineering  
University of Washington  
Seattle, WA 98195-2700  
Tel: (206) 616-2696  
Fax: (206) 543-5965  
Email: [yinhai@u.washington.edu](mailto:yinhai@u.washington.edu)

Submission Date: November 15, 2005

Word Count:  $5631 + 5 * 250 + 4 * 250 = 7881$  words

## **Abstract**

The Washington State Department of Transportation (WSDOT) has a dual-loop detection system on its Greater Seattle freeway network to provide real-time speed and vehicle classification data in a four-bin format. The dual-loop's capability of classifying vehicles makes the dual-loop detection system a potential real-time truck data source for freight movement study. However, a previous study found that the WSDOT dual-loop detectors were consistently underreporting truck volumes, whereas the single-loop detectors were consistently over counting vehicle volumes. The effectiveness of the current dual-loop algorithm was questioned.

In light of advances in controller computing capability, an improved dual-loop algorithm that can tolerate erroneous raw loop actuation signals was developed in this research. The improved dual-loop algorithm includes a noise filter and a postprocessor to screen out noise, a matching scheme to pair up on-time pulses, and an improved speed and length calculation method to increase the reliability of dual-loop outputs. When calculating speed and vehicle length, various checks are applied to test the validity of the data. If any of the checks fails, an appropriate error is flagged, but the individual vehicle data are not discarded from the total count. The data analysis conducted in this research verified the effectiveness of the noise filter and the postprocessor, the on-time pulse matching scheme, and the improved dual-loop algorithm.

Keywords: freeway traffic, inductive loop detectors, noise filter, dual-loop algorithm.

## INTRODUCTION

The Washington State Department of Transportation (WSDOT)'s freeway loop detection system is the main traffic data source for its Advanced Traffic Management Systems (ATMS) and its Advanced Traveler Information Systems (ATIS). Loop detectors are deployed as single-loop detectors, i.e., one loop per lane, and as speed traps formed by two consecutive single-loop detectors placed several meters apart in each lane. Single-loop detectors are used to measure volume and lane occupancy; while speed traps (also called dual-loop detectors, dual loops, or T loops in the state of Washington) provide two independent sets of volume and occupancy measurements and also measure speed and vehicle length.

The dual-loop's capability of measuring vehicle length makes the WSDOT freeway dual-loop detection system a potential real-time truck data source for freight movement study since truck volume estimates by basic length category can be developed from the vehicle length measurements produced by the dual-loop detectors. However, the WSDOT dual-loop detectors are consistently underreporting truck volumes, although each of the single loops that form a dual-loop detector may have counts that closely match ground-truth volumes (1). A preliminary study of the dual-loop detector detecting process found that the current WSDOT dual-loop algorithm needs to be improved (2).

Typically, a WSDOT freeway loop cabinet uses a Model 170 controller, an 8-bit 6808-based machine that was a product first released in 1975 by Motorola, Inc. The controller samples loop actuation signals at a frequency of 60 Hz. The sampled signals are then input to the dual-loop algorithm to calculate speed and vehicle length. The current WSDOT dual-loop algorithm was coded in Assembly language in order to efficiently utilize the limited hardware resources. Since the dual-loop algorithm was coded in Assembly, it is difficult to understand and update. Also perhaps because of the limited computing power of the 170 controllers, the single-loop noise filtering process was so rudimentary that a significant portion of noise could not be removed. In addition, the validity checks implemented in the current WSDOT dual-loop algorithm simply filtered out a tremendous amount of questionable loop actuation signals that could have otherwise been corrected to produce reliable speed and vehicle length estimates.

Nowadays, with advances in technologies, the computing power of controllers has been dramatically increased. The new generation of controllers is now capable of executing more involved applications. Therefore, an improved dual-loop algorithm was developed in this research to tolerate seemingly erroneous but indeed correctable raw loop actuation signals. The improved algorithm was designed to maximally filter out noise while keeping as much individual vehicle information as possible. When calculating speed and vehicle length, various checks are applied to test the validity of the data. If any of the checks fails, an appropriate error is flagged, but the individual vehicle data are not discarded from the total count.

## DUAL-LOOP ALGORITHM DESIGN

Figure 1 illustrates the responses of the two single loops when a vehicle passes a dual-loop detector. The time a vehicle occupies a loop detector, the vehicle presence-time, also known as detector on-time, can be calculated by simply subtracting the time the vehicle arrives at the detector from the time the vehicle leaves the detector. As can be seen in Figure 1, when a vehicle traverses a dual-loop detector, it sequentially occupies the paired single loops that comprise the dual-loop detector. Paired single-loop detectors' on-times ( $T_{uon}$  denotes the upstream single-loop

on-time and  $T_{don}$  denotes the downstream single-loop on-time) and the time it takes the vehicle to traverse from the upstream loop to the downstream loop, also called elapsed time ( $T_e$ ), can be calculated using the following equations:

$$T_{uon} = t_{uoff} - t_{uon} \quad (1)$$

$$T_{don} = t_{doff} - t_{don} \quad (2)$$

$$T_{e1} = t_{don} - t_{uon} \quad (3)$$

$$T_{e2} = t_{doff} - t_{uoff} \quad (4)$$

Where

- $t_{uon}$  = time when a vehicle hits the upstream loop's leading edge
- $t_{uoff}$  = time when a vehicle leaves the upstream loop's rear edge
- $t_{don}$  = time when a vehicle hits the downstream loop's leading edge
- $t_{doff}$  = time when a vehicle leaves the downstream loop's rear edge
- $T_{uon}$  = on-time at the upstream loop
- $T_{don}$  = on-time at the downstream loop
- $T_{e1}$  = elapsed time (leading edge to leading edge)
- $T_{e2}$  = elapsed time (rear edge to rear edge)

Once detector elapsed time is available, speed can be calculated straightforwardly using equation 5 or 6 by dividing the distance between the two single-loop detectors with the elapsed time. For increased reliability, speed can be calculated using equation 7 as the average of the two speeds calculated using equations 5 and 6.

$$S_1 = \frac{l_u + l_{ud}}{T_{e1}} \quad (5)$$

$$S_2 = \frac{l_d + l_{ud}}{T_{e2}} \quad (6)$$

$$S = \frac{S_1 + S_2}{2} \quad (7)$$

Where

- $l_u$  = length of the upstream loop
- $l_d$  = length of the downstream loop
- $l_{ud}$  = distance between the upstream and downstream loops
- $S_1$  = speed calculated using the length of the upstream loop and the elapsed time from the leading edge of the upstream loop to that of the downstream loop
- $S_2$  = speed calculated using the length of the downstream loop and the elapsed time from the rear edge of the upstream loop to that of the downstream loop
- $S$  = average of  $S_1$  and  $S_2$

Vehicle speed and single-loop on-times are then used to calculate vehicle length using either of the following equations:

$$l_{vu} = T_{uon} \times S - l_u \quad (8)$$

$$l_{vd} = T_{don} \times S - l_d \quad (9)$$

Where

$l_{vu}$  = vehicle length calculated using the upstream loop's on-time and length

$l_{vd}$  = vehicle length calculated using the downstream loop's on-time and length

When both of the loops work properly (i.e., either of the two calculated lengths can be used to estimate vehicle length), the average of the two calculated lengths (equation 10) can be used as the length of the target vehicle for increased reliability.

$$l_v = \frac{l_{vu} + l_{vd}}{2} \quad (10)$$

Where

$l_v$  = average of  $l_{vu}$  and  $l_{vd}$

As discussed above, the dual-loop algorithm itself is very straightforward when the dual-loop detectors are working properly. However, inductive loop detectors are subject to malfunctions (3) and the malfunctioning loops generate erroneous raw loop actuation signals, which lead to unreliable speed and length measurements if the loop algorithm fails to detect and correct those erroneous raw signals. Therefore, in the improved dual-loop algorithm, various validity checks are conducted to flag erroneous loop actuation signals along with the calculated speed and vehicle length.

## NOISE FILTER AND POSTPROCESSOR

### Introduction to the Noise Filter and Postprocessor

Almost all electronic signals are subject to interference from noise. The loop actuation signals are digital signals, which have two states – “on” and “off.” These two states indicate whether or not the loop is being occupied; where “on” means occupied and “off” means not occupied. The duration of an “on” state is called a loop's on-time. The “on” and “off” states can be combined to form an infinite number of “on” and “off” sequences, but not all of them are valid signal sequences. Therefore, before using the loop actuation signals to calculate speed and vehicle length information, it is necessary to apply a noise filter to filter out random positive (loop detector is in “on” state when there is no vehicle occupying the loop) or negative (loop detector stays in “off” state when there is a vehicle occupying the loop) false alarm noise or pulse breaks. A postprocessor should also be applied to catch any false alarm signals induced or undetected by the noise filter.

The minimum on-time threshold value is 75 milliseconds derived by dividing the minimum effective vehicle length (sum of the minimum vehicle length of five feet (1.5 meters) and the loop length which is six feet (1.8 meters) for the WSDOT loop detection system with the maximum vehicle speed which is considered to be 100 miles per hour (161 kilometers per hour) on the freeways in the Greater Seattle Area. Because the controller's sampling interval is 16 milliseconds, 75 milliseconds leads to a signal sequence of five 16-milliseconds in length. Therefore, five 16-milliseconds is a proper window size for the noise filter. For convenience, "0" is used to represent the "on" state, and "1" is used to represent the "off" state. The noise filter can then be presented as a 5-digit digital signal sequence. The noise filter developed in this research is pictorially represented as a Karnaugh map (4) in Table 1.

The Karnaugh map, also known as a Veitch diagram, is a tool to facilitate management of Boolean algebraic expressions. In the Karnaugh map, AB represents the first two digits of the 5-digit input signal sequence, DE represents the last two digits of the 5-digit input signal sequence, and C represents the digit in the middle that needs to be determined by the noise filter. In total, there are 32 different 5-digit input signal sequences, which exhaust all possible 5-digit combinations of zeros and ones.

Because the improved dual-loop algorithm aims at detecting as many vehicles as possible, the objective of this noise filter is to maximally filter out noise, while still keeping as many on-time pulses as possible for each of the two single loops. Although some positive false alarm signals may be introduced in this process, they will be filtered out during the matching process in the dual-loop algorithm.

The noise filter screens out positive or negative single-digit false alarm noise by flipping the digit from its current state to its opposite state. In the noise filter a "0" in the middle of the input signal sequence will not be flipped to "1" unless the digit is preceded by two "1"s and succeeded by two "1"s; a "1" in the middle of the input sequence will not be flipped to "0" if the digit is preceded by two "1"s, or succeeded by two "1"s, or preceded by a "1" and succeeded by a "1." This will retain as many on-time pulses as possible during the noise-filtering process. The digits that have been flipped are underlined in the Karnaugh map,

After being processed by the noise filter, the loop actuation signals will be sent to a postprocessor to filter out any positive or negative false alarm signal shorter than the on-time or off-time threshold value. The output loop actuation signals will then be used to calculate speed and vehicle-length information.

## **Illustration of the Noise Filter and Postprocessor**

All the possible combinations of zeros and ones can be classified into six groups, each of which can be reduced to and represented by a general signal sequence. The noise filter and postprocessor are applied to each of these six representative signals to test their effectiveness. These processes are illustrated in Figure 2. In this figure, the noise recognized by the noise filter in the original signals is labeled for each group.

The original signal sequence in Case 1 represents those signal sequences with random single-digit noise occurring intermittently when a loop detector is in the "on" state. The noise filter easily filters out this type of noise, and the broken on-time pulse is reconstructed.

The original signal sequence in Case 2 represents those signal sequences with random single-digit and 2-digit noise occurring through an on-time pulse and the noise occurring right before the loop detectors change their state from "on" to "off." As can be seen in Case 2 of

Figure 2, the negative false alarm noise occurring in the “on” state and the single-digit positive false alarm noise occurring in the “off” state, are filtered out by the noise filter. However, the noise filter mistakenly identifies the last digit of the on-time pulse as a single-digit positive false alarm noise because, as previously stated, if a “0” digit is preceded and succeeded by two or more “1”s, this digit is treated as a positive false alarm noise and flipped from “0” to “1”. In Case 2, the broken on-time pulse in the original signal sequence is reconstructed at the price of sacrificing its last on-time digit.

The original signal sequence in Case 3 represents those signal sequences with random single-digit noise occurring when a loop detector is in the “off” state. The random noise occurs in a way such that it can “fool” the noise filter into introducing a positive false alarm signal that is detected by the subsequent postprocessor. As shown in Case 3 of Figure 2, positive false alarm noise occurs in the “off” state in the original signal sequence. When this signal sequence is being filtered, the second and the third positive false alarm noise is ignored; instead, the digit between these two positive false alarm noise is mistakenly identified as a single-digit negative false alarm noise by the noise filter. After being processed by the noise filter, the first positive false alarm noise is filtered out. The mistakenly identified negative false alarm noise is flipped from its “off” state to the “on” state, which bridges the second and the third positive false alarm noise to form a short on-time pulse of four samples long. Since this short on-time pulse is shorter than the minimum on-time threshold value, it is filtered out as a positive false alarm signal by the subsequent postprocessor. In Case 3, the noise is successfully filtered out by the joint effort of the noise filter and the postprocessor.

The original signal sequence in Case 4 represents those signal sequences with random noise occurring right before and after a loop detector changes its state from “on” to “off.” As shown in Case 4 of Figure 2, in the original signal sequence, five negative false alarm noise occurs, breaking part of the on-time pulse into pieces right before the loop detector changes its state, and then three positive false alarm noise occurs right after the loop detector changes its state from “on” to “off.” After being processed by the noise filter, these two types of noise are filtered out.

The original signal sequence in Case 5 represents those signal sequences with multi-digit noise immune to the noise filter but sensitive to the postprocessor, i.e., the noise goes through the noise filter undetected, but is detected by the postprocessor. As can be seen in Case 5 of Figure 2, the three 3-digit negative false alarm noise is not detected by the noise filter but filtered out by the postprocessor.

The original signal sequence in Case 6 represents those signal sequences with random noise occurring in such a way that it can “fool” the noise filter into introducing a negative false alarm signal, which is later detected by the postprocessor. Shown in Case 6 of Figure 2, the noise filter recognizes six out of seven negative false alarm noise. It misses the sixth because it mistakenly identifies the single-digit between the fifth and the sixth false alarm noise as a single-digit positive false alarm noise. After being processed by the noise filter, part of the on-time pulse is recovered. However, the noise filter induces a negative false alarm signal by flipping the mistakenly identified single-digit positive false alarm noise from “on” to “off”, which then merges with the sixth negative false alarm noise to form a negative false alarm signal. This negative false alarm signal is removed by the postprocessor. In Case 6, the noise is successfully filtered out by the joint effort of the noise filter and the postprocessor.

## PAIRED ON-TIME MATCHING

### Introduction to the Single Loop On-Time Pulse Matching Rules

After the noise filtering and post processing process, the upstream loop (called M loop hereafter per the WSDOT naming convention) and the downstream loop (S loop hereafter per the WSDOT naming convention) signal sequences are processed simultaneously to match the M loop and S loop on-time pairs. If a dual-loop detector is working properly and the vehicle does not switch to another lane between M and S loops, the on-time pulse generated by the M loop should have a matching pulse generated by the S loop. Therefore, for each on-time pulse in the M loop signal sequence, the paired on-time matching algorithm searches for its best matching pulse in the S loop signal sequence. Seven rules are applied during this matching process.

Rule 1: When a vehicle traverses a dual-loop detector, it first occupies the M loop and then the S loop, so the S loop on-time matching pulse should occur after the M loop on-time pulse. The S loop on-time matching pulse should also occur before the next M loop on-time pulse due to the short distance between the M and S loops and, typically, the current vehicle hits the S loop before the following vehicle hits the M loop.

Rule 2: The elapsed time from the beginning of the M loop on-time pulse to the beginning of its matching S loop on-time pulse should be longer than the minimum elapsed time threshold and shorter than the maximum elapsed time threshold.

Rule 3: If the S loop on-time pulse satisfies Rule 1 but does not meet the criterion stated in Rule 2, an error indicating that the elapsed time is shorter than the minimum threshold or greater than the maximum threshold is flagged. If more than one S loop on-time pulse satisfy Rule 1 but not Rule 2, the first S loop on-time pulse is selected.

Rule 4: The M loop on-time pulse that does not have any matching pulse in the S loop signal sequence is filtered out as a positive false alarm signal.

Rule 5: The S loop on-time pulse that does not have a matching pulse in the M loop signal sequence is filtered out as a positive false alarm signal.

Rule 6: If one S loop on-time pulse satisfies Rule 2 for two or more M loop on-time pulses, the M loop on-time pulse that immediately precedes the S loop on-time pulse in the M loop on-time sequence is chosen to form a matching pair with the S loop on-time pulse.

Rule 7: If two or more S loop on-time pulses satisfy Rule 2 for an M loop on-time pulse, the S loop on-time pulse that immediately succeeds the M loop on-time pulse is chosen to form a matching pair with the M loop on-time pulse.

## Illustration of the M and S Loop On-Time Pulse Matching Rules

Typical positive false alarm pulses that could occur when matching for the M loop and S loop on-time pulse pairs can be classified into five cases. The application of the seven rules is illustrated in Figure 3 by processing these five cases.

In Case 1, a positive false alarm occurs in the S loop signal sequence and must be skipped during the matching process. The rules employed in this case include Rule 1, Rule 2, and Rule 5. In this case, the S loop on-time pulse  $S_j$  succeeds the M loop on-time pulse  $M_i$ , and the elapsed time,  $T_e(i, j)$ , from  $M_i$  to  $S_j$  is greater than the minimum elapsed time threshold and smaller than the maximum elapsed time threshold. Therefore, according to Rule 1 and Rule 2,  $S_j$  is matched to  $M_i$ . The on-time pulse  $S_{j-1}$ , however, cannot be matched to any M loop on-time pulse. The M loop on-time pulse,  $M_{i-1}$ , which immediately precedes  $S_{j-1}$ , forms an on-time pair with  $S_{j-2}$ . So, according to Rule 5,  $S_{j-1}$  is filtered out as a positive false alarm.

In Case 2, a positive false alarm occurs in the M loop signal and needs to be skipped or screened out during the matching process. The rules employed in this case include Rule 1, Rule 2, and Rule 4. In this case, the S loop on-time pulse  $S_j$  succeeds the M loop on-time pulse  $M_i$ , and the elapsed time,  $T_e(i, j)$ , from  $M_i$  to  $S_j$  falls in the valid range for elapsed time. So, according to Rule 1 and Rule 2,  $S_j$  is matched to  $M_i$ . The on-time pulse  $M_{i-1}$ , however, cannot be matched to any S loop on-time pulse. The S loop on-time pulse,  $S_j$ , which immediately succeeds  $M_{i-1}$ , forms an on-time pair with  $M_i$ . In addition, the elapsed time from  $M_{i-1}$  to  $S_j$  is greater than the maximum elapsed time threshold. Therefore, according to Rule 4,  $M_{i-1}$  is filtered out as a positive false alarm.

In Case 3, the elapsed time between the M loop on-time pulse and its matching S loop on-time pulse is greater than the maximum elapsed time threshold. The rules employed in this case include Rule 1 and Rule 3. In this case,  $S_j$  immediately succeeds  $M_i$  and is the only S loop on-time pulse that can be matched to  $M_i$ . Although the elapsed time between  $M_i$  and  $S_j$  is greater than the maximum elapsed time threshold, according to Rule 1 and Rule 3,  $M_i$  and  $S_j$  form a pair of on-time pulses.

In Case 4, an S loop on-time pulse can be matched to either of two M loop on-time pulses. This S loop on-time pulse forms an on-time pair with the M loop on-time pulse that immediately precedes it. The rules employed in this case include Rule 1, Rule 2, and Rule 6. In this case, according to Rule 1 and Rule 2,  $S_j$  can be matched to either of two M loop on-time pulses,  $M_{i-1}$  and  $M_i$ , because the elapsed time from either of the two M loop on-time pulses to  $S_j$  falls in the valid range. Then, according to Rule 6,  $S_j$  is matched to  $M_i$  because  $M_i$  immediately precedes  $S_j$ .

In Case 5, an M loop on-time pulse can be matched to two S loop on-time pulses. This M loop on-time pulse forms an on-time pair with the S loop on-time pulse that immediately succeeds it. The rules employed in this case include Rule 1, Rule 2, and Rule 7. In this case, according to Rule 1 and Rule 2,  $M_j$  can be matched to either  $S_j$  or  $S_{j+1}$ , because the elapsed time from  $M_j$  to either of the two S loop on-time pulses falls in the valid range. Then, according to Rule 7,  $S_j$  is matched to  $M_i$  because  $S_j$  immediately succeeds  $M_i$ .

## INDIVIDUAL VEHICLE SPEED CALCULATION

After the M loop and S loop on-time pulses are matched, the matched on-time pulse pairs are used to calculate speed. As shown in Figure 1, there are two elapsed times,  $T_{e1}$  and  $T_{e2}$ , for each on-time pulse pair, so in the improved dual-loop algorithm, the validity of  $T_{e1}$  and  $T_{e2}$  is checked.

If either or both falls outside of the valid range or the relative difference between the two ( $\Delta T_e$ ) is greater than the threshold value ( $\Delta T_{e\_threshold}$ ) the speed of the preceding vehicle will be used in the calculation considering that, under un-congested traffic conditions, the speeds of two consecutive vehicles running on the freeway do not differ considerably. The process employed to calculate speeds is illustrated in the flow chart in Figure 4. Here, the vehicle for which speed and length data are calculated is represented by index  $i$ , and the vehicle that precedes the current one is represented by index  $i-1$ .

In the current WSDOT dual-loop algorithm,  $\Delta T_{e\_threshold}$  equals 10%. For consistency, 10% is also used in this research. For each pair of M and S loop on-time pulses, elapsed times  $T_{e1}$  and  $T_{e2}$  are compared to the elapsed time threshold values,  $T_{emin}$  and  $T_{emax}$ , and the difference between them,  $\Delta T_e$ , is compared to  $\Delta T_{e\_threshold}$ . If any of the parameters falls outside the valid range, an error is flagged. There are four possible cases:

In Case 1, both elapsed times fall within the valid range, i.e.,  $T_{emin} < T_{e1} < T_{emax}$  and  $T_{emin} < T_{e2} < T_{emax}$ . In addition, if  $\Delta T_e$  is smaller than  $\Delta T_{e\_threshold}$ , both elapsed times are used to calculate speeds. If  $\Delta T_e$  is larger than  $\Delta T_{e\_threshold}$ , the speed of the vehicle (if there is any) which precedes the current one is used to derive the elapsed time  $T_e'$ . Then, of the two elapsed times,  $T_{e1}$  and  $T_{e2}$ , the one that is closer in value to  $T_e'$  will be used for the speed calculation. In this case, an error is flagged. If  $\Delta T_e$  is larger than  $\Delta T_{e\_threshold}$  and the current vehicle is the first one for which speed is to be calculated, both  $T_{e1}$  and  $T_{e2}$  are used to calculate speeds.

In Case 2, the first elapsed time falls within the valid range, the second elapsed time falls outside the valid range, i.e.,  $T_{emin} < T_{e1} < T_{emax}$  and  $T_{e2} < T_{emin}$  or  $T_{e2} > T_{emax}$ . If there is no vehicle preceding the current one, only  $T_{e1}$  is used for the speed calculation because  $T_{e2}$  is invalid. If there is a vehicle preceding the current one, the speed of that vehicle is used to derive the elapsed time  $T_e'$ . If the difference between  $T_e'$  and  $T_{e1}$  is greater than  $\Delta T_{e\_threshold}$ , only  $T_{e1}$  is used for the speed calculation. Otherwise, the speed of the preceding vehicle is also used in the speed calculation for the current vehicle for improved reliability. The reasoning behind this is that the speeds of successive vehicles running on the freeway under un-congested traffic conditions do not differ considerably.

In Case 3, the first elapsed time falls outside the valid range, the second elapsed time falls inside the valid range, i.e.,  $T_{e1} < T_{emin}$  or  $T_{e1} > T_{emax}$  and  $T_{emin} < T_{e2} < T_{emax}$ . Case 3 is similar to Case 2, so the speed calculation method in Case 2 is also applied in Case 3 except that  $T_{e2}$  rather than  $T_{e1}$  is used in speed calculation.

In Case 4, neither of the two elapsed times falls inside the valid range, i.e.,  $T_{emin} > T_{e1}$  or  $T_{e1} > T_{emax}$ , and  $T_{emin} > T_{e2}$  or  $T_{e2} > T_{emax}$ . In this case, the speed of the current vehicle is set equal to the speed of the preceding vehicle because the speeds of successive vehicles running on the freeway under un-congested traffic conditions should not differ considerably. If the current vehicle is the first one detected in the data collection, that vehicle's speed is set to zero.

## INDIVIDUAL VEHICLE LENGTH CALCULATION

The calculated speed is next used to calculate individual vehicle length. In the improved dual-loop algorithm, the validity of the two on-times ( $T_{uon}$  and  $T_{don}$ ) is checked. If either or both fall outside of the valid range, or, if the difference between the two on-times ( $\Delta T_{on}$ ) is greater than the on-time difference threshold value ( $\Delta T_{on\_threshold}$ ), errors will be flagged. The process employed to calculate vehicle lengths is illustrated in the flow chart in Figure 5.

Each pair of  $T_{uon}$  and  $T_{don}$  are compared to the on-time threshold values,  $T_{onmin}$  and  $T_{onmax}$ , and the difference between them,  $\Delta T_{on}$ , is compared to  $\Delta T_{on\_threshold}$ , which is 10% in this research to be consistent with the current WSDOT dual-loop algorithm. There are four possible cases:

In Case 1, both on-times fall within the valid range, i.e.,  $T_{onmin} < T_{uon} < T_{onmax}$  and  $T_{onmin} < T_{don} < T_{onmax}$ , so both are used for vehicle length calculation. If  $\Delta T_{on}$  is greater than  $\Delta T_{on\_threshold}$ , an error is flagged.

In Case 2,  $T_{uon}$  falls within the valid range but  $T_{don}$  falls outside the valid range, i.e.,  $T_{onmin} < T_{uon} < T_{onmax}$  and  $T_{don} < T_{onmin}$  or  $T_{don} > T_{onmax}$ . In this case, only  $T_{uon}$  is used for vehicle length calculation with an error flagged.

In Case 3,  $T_{don}$  falls within the valid range but  $T_{uon}$  falls outside the valid range, i.e.,  $T_{uon} < T_{onmin}$  or  $T_{uon} > T_{onmax}$ , and  $T_{onmin} < T_{don} < T_{onmax}$ . In this case, only  $T_{don}$  is used for vehicle length calculation with an error flagged.

In Case 4, neither of the two on-times falls within the valid range, i.e.,  $T_{uon} < T_{onmin}$  or  $T_{uon} > T_{onmax}$ , and  $T_{don} < T_{onmin}$  or  $T_{don} > T_{onmax}$ . In this case, both on-times are used for length calculation with an error flagged.

## EFFECTIVENESS OF THE IMPROVED DUAL-LOOP ALGORITHM

The effectiveness of the improved dual-loop algorithm was evaluated in this research by comparing traffic data calculated using the improved dual-loop algorithm to those calculated using the current WSDOT dual-loop algorithm and video ground-truth data. Data compared included volume and vehicle classification. The evaluation results are summarized in Table 2, Table 3, and Table 4, and discussed as follows:

- Shown in Table 2, the single-loop volume calculated using the improved dual-loop algorithm was almost equal (less than 0.32 percent difference) to the video-ground-truth volume for each of the travel lanes where data were collected. The single-loop volume calculated using the current WSDOT dual-loop algorithm, in contrast, was consistently higher (up to 4.19 percent) than the video-ground-truth volume for each of the travel lanes.

Since the current WSDOT dual-loop algorithm has only a rudimentary noise filtering process, a tremendous amount of noise is overlooked resulting in the single-loop over-count problem. With the implementation of the noise filter and the postprocessor in the improved dual-loop algorithm, the single-loop over-count problem should be significantly alleviated. The comparison results above prove the effectiveness of the noise filter and postprocessor.

- As can be seen in Table 3, the number of passing vehicles calculated by the improved dual-loop algorithm closely (less than 0.08 percent difference) matched the corresponding video-ground-truth data for each of the travel lanes where data were collected. The dual-loop data calculated using the current WSDOT dual-loop algorithm, in contrast, consistently undercounted (up to 9.22 percent) vehicles.

The improved dual-loop algorithm conducts various checks to test the validity of individual vehicle data. If any of the checks fails, an appropriate error is flagged. The current WSDOT dual-loop algorithm also conducts some validity checks and flags errors, but simply discards data with error flags. Conversely, the improved dual-loop algorithm

keeps the individual vehicle data with error flags in the total count to alleviate the dual-loop undercount problem. The comparison results above prove the effectiveness of the validity checks.

- Shown in Table 4, the improved dual-loop algorithm correctly classified significantly more vehicles than the WSDOT dual-loop algorithm, especially, for the case of large vehicles, where the improved dual-loop algorithm correctly classified up to 33 percent more vehicles.

The improved dual-loop algorithm applies seven rules when matching M loop and S loop on-time pulses for individual vehicle speed calculation. The improved dual-loop algorithm also takes into account the fact that under non-forced-flow traffic conditions, speeds of vehicles in a platoon should not differ significantly. So if the calculated speed of the current vehicle is problematic, the speed of the preceding vehicle is used to adjust the calculated speed. The current WSDOT dual-loop algorithm does not have this feature. The on-time matching process, the speed adjustment, and the features discussed above should improve the accuracy of the dual-loop classification data. The comparison results above prove the effectiveness of these features.

## CONCLUSIONS

In this research, an improved dual-loop algorithm that can tolerate seemingly erroneous but indeed correctable raw loop actuation signals was developed. This newly developed algorithm was designed to maximally filter out noise while keeping as much individual vehicle information as possible despite some unreliability in the raw loop actuation signals. The improved dual-loop algorithm reduced the single-loop over count rate, increased the dual-loop detection rate, and improved the accuracy of vehicle classification data.

In conclusion, the advantages of this improved dual-loop algorithm over the current WSDOT dual-loop algorithm include the following:

- Inductive raw loop actuation signals subject to noise. The current WSDOT dual-loop algorithm has a rudimentary noise filter, which does not effectively filter out noise, resulting in the single-loop over count problem. The noise filter and postprocessor implemented in the improved dual-loop algorithm effectively filter out noise and successfully reduce the single-loop over count rate.
- Inductive loop measurements subject to errors induced by a variety of causes such as incorrect system software settings or simple hardware failure. The current WSDOT dual-loop algorithm conducts some validity checks to flag errors and then simply discards the data with error flags resulting in the dual-loop undercount problem. The improved dual-loop algorithm, conversely, employs a set of thorough validity checks and keeps the individual vehicle data with error flags in the total count to increase the dual-loop detection rate. The on-time matching rules and the speed adjustment feature applied in the improved dual-loop algorithm also improve the accuracy of dual-loop outputs.

**References**

1. Zhang, X., Y. Wang, and N. L. Nihan. Investigating Dual-Loop Errors Using Video Ground-Truth Data. *Proc. ITS America 2003 Annual Meeting*, Minneapolis, Minnesota, 2003.
2. Zhang, X., N. L. Nihan, Y. Wang. An Improved Dual-Loop Detection System for Collecting Real-Time Truck Data. *Preprint CD-ROM*, Transportation Research Board, Washington, DC, 2005.
3. Traffic Detector Handbook (Second Edition). Institute of Transportation Engineers. Washington, DC, 1997.
4. Holdsworth, B., and R. C. Woods. *Digital Logic Design*. Oxford, Newnes, 2002. ISBN: 0750645822.
5. Zhang, X., Y. Wang, and N. L. Nihan. Improved System for Collecting Real-Time Truck Data from Dual-Loop Detectors. Publication TNW2005-02. FHWA, U.S. Department of Transportation, 2005.

**Acknowledgement**

This work was funded by Transportation Northwest (TransNow), U.S. Department of Transportation University Center for Federal Region 10. The authors thank Mark E. Hallenbeck, who provided direction to some of the issues and reviewed the paper. The authors thank Mark Morse and Lanping Xu of the Washington State Department of Transportation, who provided assistance in software testing and data collection. The authors also thank Nathan Miller, who helped search and purchase interfacing hardware for developing the DEDAC system.

**List of Figures and Tables**

Figure 1. Illustration of a vehicle passing over a dual-loop detector

Figure 2. Illustration of the noise filter and postprocessor

Figure 3. Illustration of M loop and S loop on-time matching

Figure 4. Speed calculation flow chart

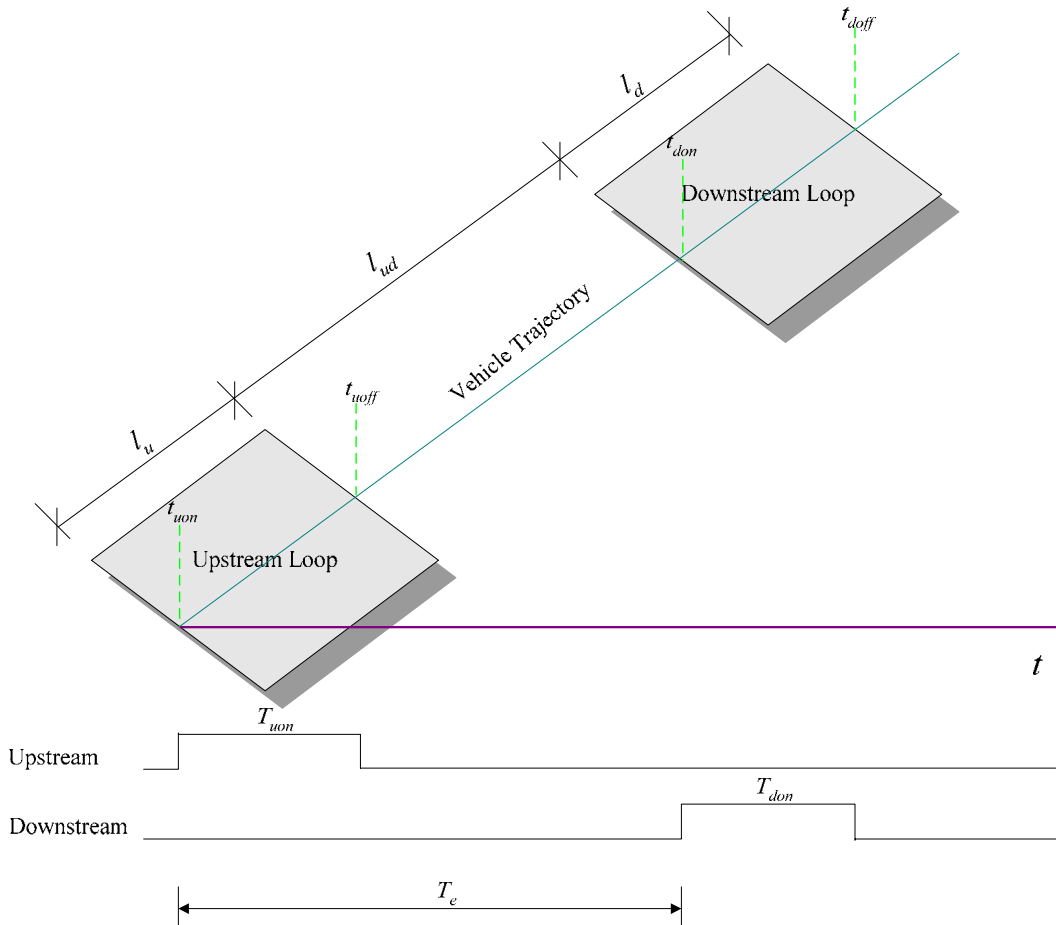
Figure 5. Flow chart for vehicle length calculation

Table 1. The Noise Filter in the Format of Karnaugh Map

Table 2. One-Hour WSDOT-Algorithm-Based and Proposed-Algorithm-Based Single-Loop and Video-Ground-Truth Volumes

Table 3. One-Hour Proposed-Algorithm-Based and WSDOT-Algorithm-Based Dual-Loop Volume Undercount Rates

Table 4. One-Hour Video-Based, Proposed-Algorithm-Based, and WSDOT-Algorithm-Based Vehicle Classification Data



Where

- $t_{uon}$  = time when a vehicle hits the upstream loop's leading edge
- $t_{uoff}$  = time when a vehicle leaves the upstream loop's rear edge
- $t_{don}$  = time when a vehicle hits the downstream loop's leading edge
- $t_{doff}$  = time when a vehicle leaves the downstream loop's rear edge
- $l_u$  = length of the upstream loop
- $l_d$  = length of the downstream loop
- $l_{ud}$  = distance between the upstream and downstream single loops measured from the rear edge of the upstream loop to the leading edge of the downstream loop
- $T_{uon}$  = on-time at the upstream loop
- $T_{don}$  = on-time at the downstream loop
- $T_{e1}$  = elapsed time (leading edge to leading edge)
- $T_{e2}$  = elapsed time (rear edge to rear edge)

Figure 1. Illustration of a vehicle passing over a dual-loop detector

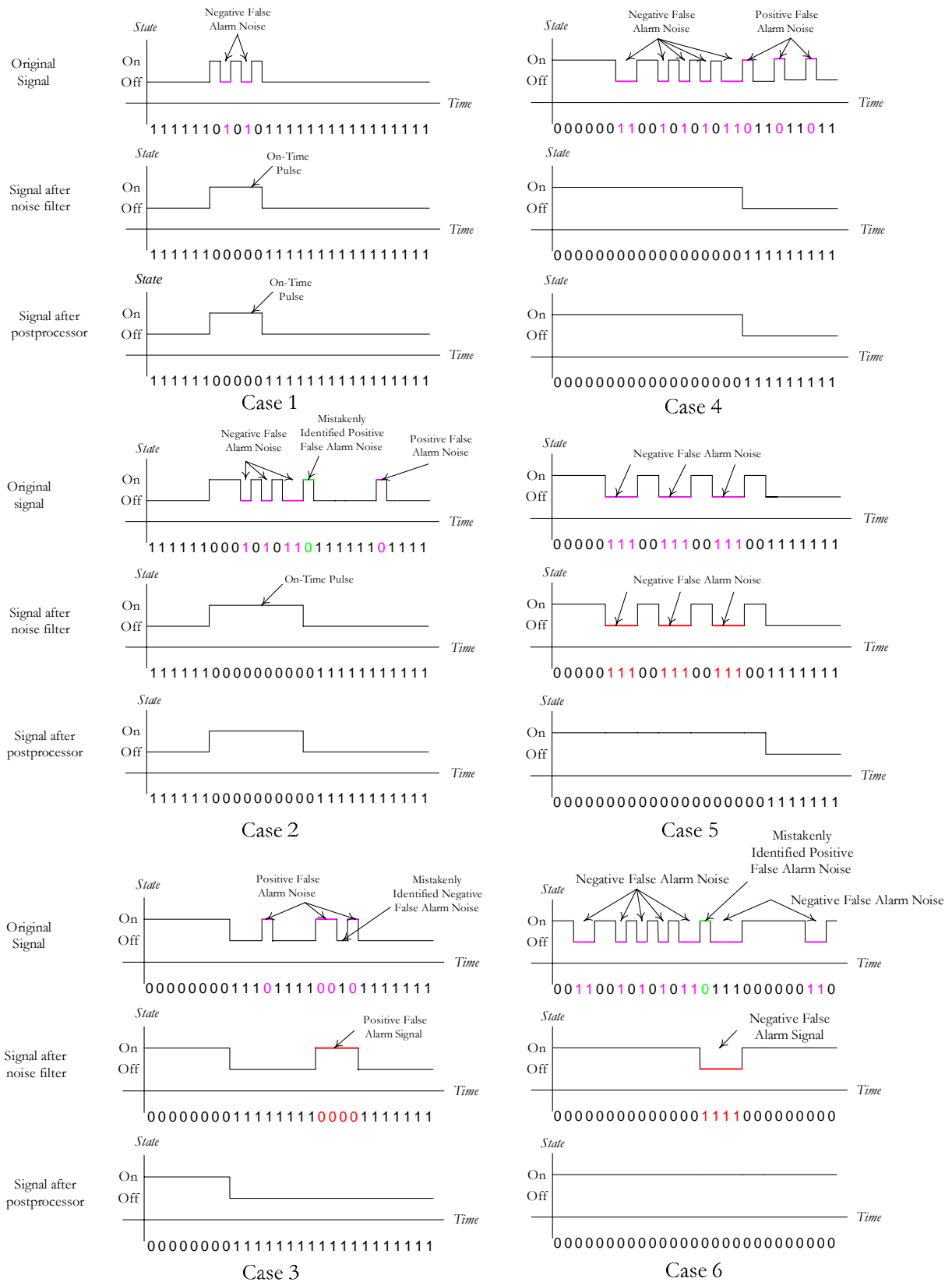


Figure 2. Illustration of the noise filter and postprocessor

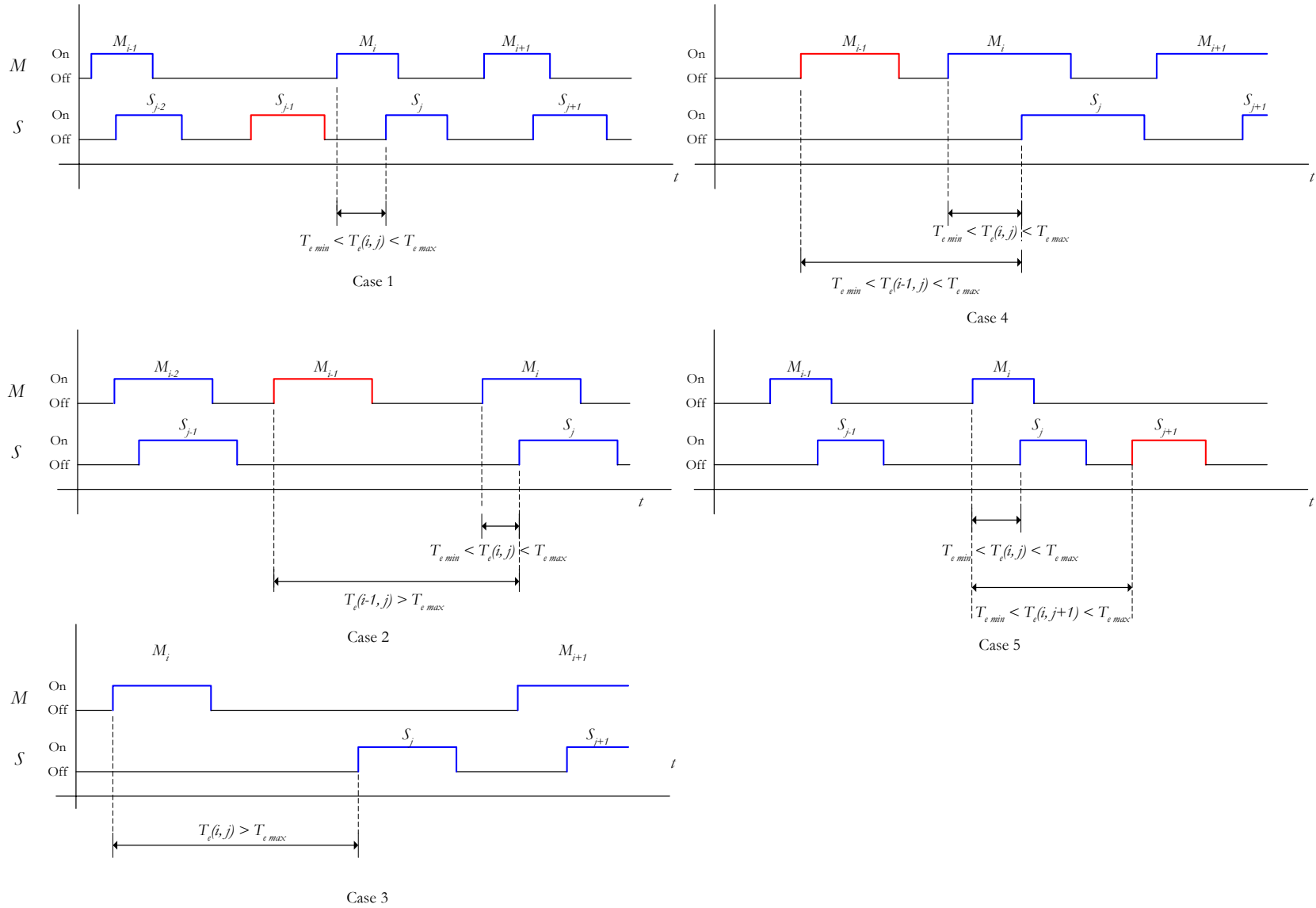


Figure 3. Illustration of M loop and S loop on-time matching

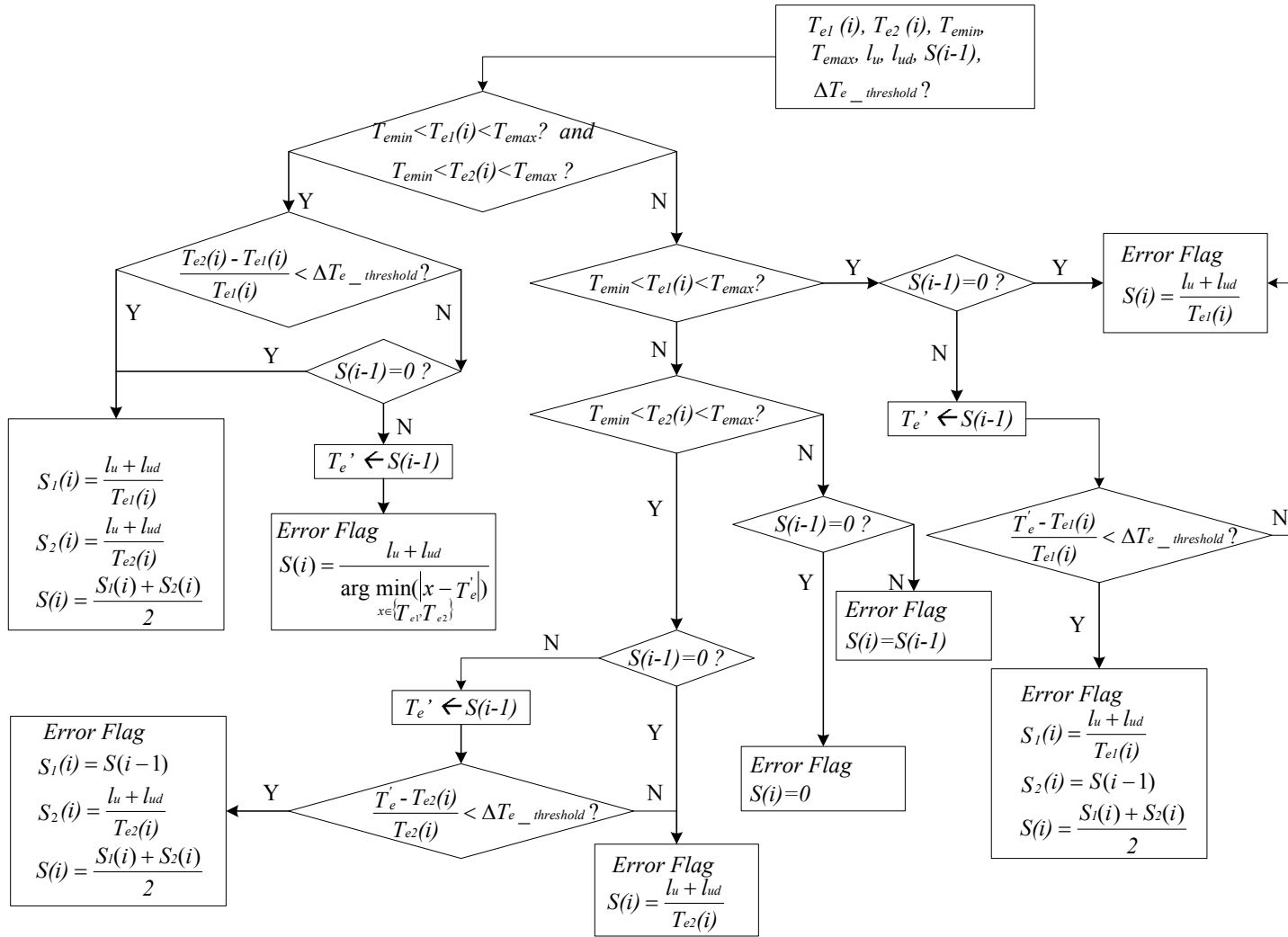


Figure 4. Flow chart for speed calculation

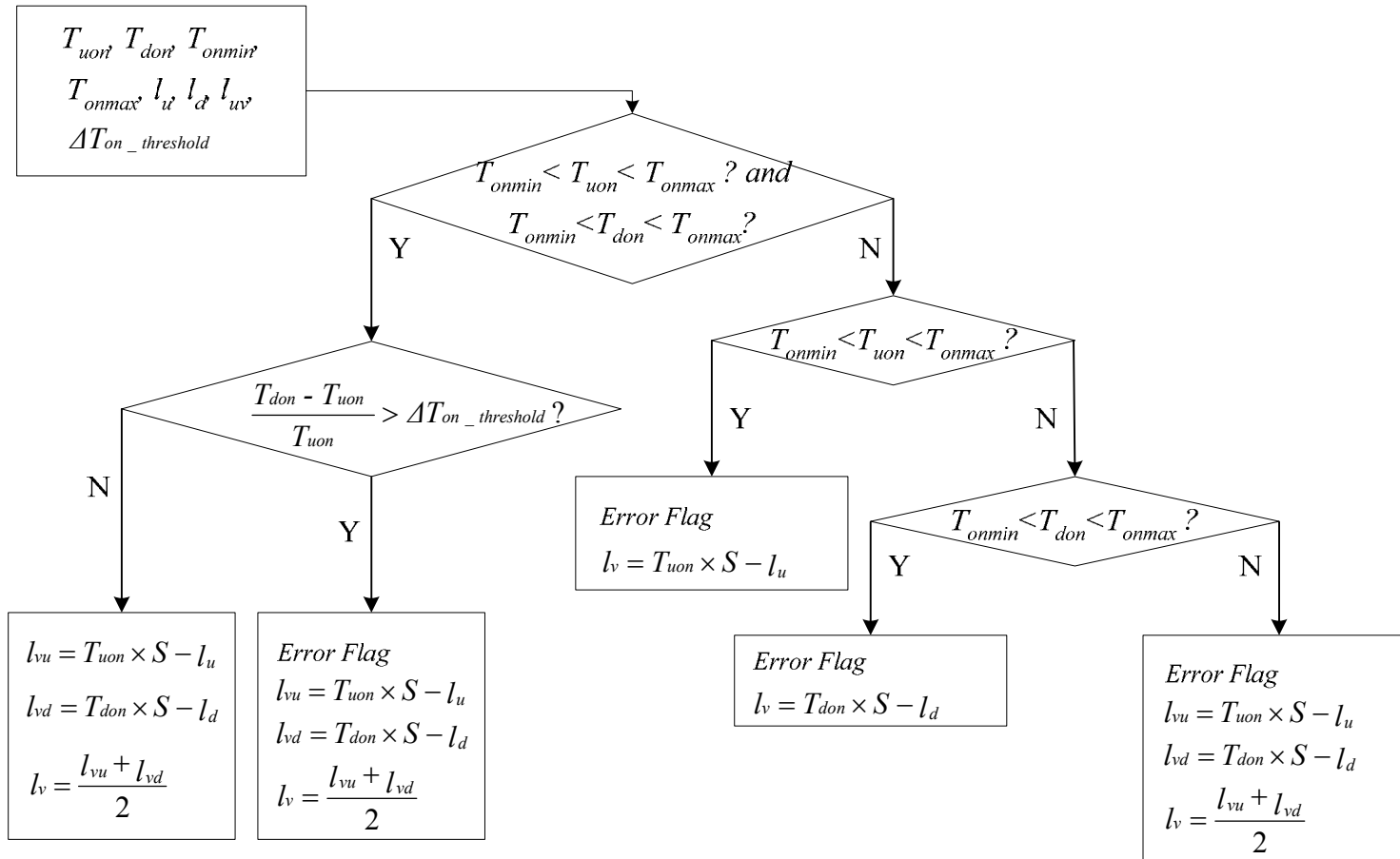


Figure 5. Flow chart for vehicle length calculation

Table 1. The Noise Filter in the Format of Karnaugh Map

AB \ CDE	000	001	010	011	100	101	110	111
00	0	0	0	0	<u>0</u>	<u>0</u>	<u>0</u>	1
01	0	0	0	0	<u>0</u>	<u>0</u>	1	1
10	0	0	0	0	<u>0</u>	<u>0</u>	<u>0</u>	1
11	0	0	0	<u>1</u>	1	1	1	1

Table 2. One-Hour WSDOT-Algorithm-Based and Proposed-Algorithm-Based Single-Loop and Video-Ground-Truth Volumes

Lane No	TDAD-Based				Event-Data-Based				Video-Ground-Truth $V_V$	Over-Count Rate Reduction	
	$V_M$	$\frac{V_M - V_V}{V_V}$	$V_S$	$\frac{V_S - V_V}{V_V}$	$V_{EM}$	$\frac{V_{EM} - V_V}{V_V}$	$V_{ES}$	$\frac{V_{ES} - V_V}{V_V}$		$\frac{V_M - V_{EM}}{V_V}$	$\frac{V_S - V_{ES}}{V_V}$
Lane 1	1308	3.65%	1315	4.19%	1263	0.08%	1266	0.32%	1262	3.57%	3.87%
Lane 2	1644	3.85%	1654	4.49%	1583	0.00%	1585	0.13%	1583	3.85%	4.36%
Lane HOV	350	4.79%	347	3.89%	334	0.00%	334	0.00%	334	4.79%	3.89%

$V_M$  = WSDOT-algorithm-based M loop volume

$V_S$  = WSDOT-algorithm-based S loop volume

$V_{EM}$  = Proposed-algorithm-based M loop volume

$V_{ES}$  = Proposed-algorithm-based S loop volume

$V_V$  = Video-ground-truth volume

Table 3. One-Hour Proposed-Algorithm-Based and WSDOT-Algorithm-Based Dual-Loop Volume Undercount Rates

Lane No	$V_T$	$V_{ET}$	$V_V$	$\frac{V_T - V_V}{V_V}$	$\frac{V_{ET} - V_V}{V_V}$
Lane 1	1169	1263	1262	-7.37%	0.08%
Lane 2	1437	1583	1583	-9.22%	0.00%
Lane HOV	320	334	334	-4.19%	0.00%

$V_T$  = Dual-loop volume output by the current WSDOT dual-loop algorithm

$V_{ET}$  = Dual-loop volume output by the proposed dual-loop algorithm

$V_V$  = Video-ground-truth volume

Table 4. One-Hour Video-Based, Proposed-Algorithm-Based, and WSDOT-Algorithm-Based Vehicle Classification Data

Bin No.	Lane 1					Lane 2					Lane HOV				
	$V_V$	$V_E$	$V_T$	$\frac{V_E}{V_V}$	$\frac{V_T}{V_V}$	$V_V$	$V_E$	$V_T$	$\frac{V_E}{V_V}$	$\frac{V_T}{V_V}$	$V_V$	$V_E$	$V_T$	$\frac{V_E}{V_V}$	$\frac{V_T}{V_V}$
Bin1	973	973	914	100%	94%	1463	1463	1326	100%	91%	328	328	306	100%	93%
Bin2	39	39	30	100%	77%	12	12	7	100%	58%	0	0	0	----	----
Bin3	44	44	35	100%	80%	20	20	14	100%	70%	0	0	0	----	----
Bin4	88	87	73	99%	83%	24	24	16	100%	67%	0	0	0	----	----
Subtotal	1144	1143	1052	99.9%	92.0%	1519	1519	1363	100.0%	89.7%	328	328	306	100%	93.3%

$V_V$  = Number of vehicles that were classified into bins\* when processing the videotape

$V_E$  = Number of vehicles that were classified into the same bins by the proposed dual-loop algorithm and by processing the videotape

$V_T$  = Number of vehicles that were classified into the same bins by the current WSDOT dual-loop algorithm and by processing the videotape

\* In the current WSDOT system, vehicles are classified according to their lengths by assigning each identified vehicle to one of four bins: (a) Bin 1 - PCs and smaller vehicles (length 26 ft or less); (b) Bin 2 - small trucks, etc. (26 ft to 39 ft); (c) Bin 3 - larger trucks and buses (39 ft to 65 ft); and (d) Bin 4 - largest trucks and articulated buses (length greater than 65 ft).