Overview of the Kadence Real-Time Adaptive Control System

FHWA SBIR DTRT57-10-C-10044

Douglas Gettman, Ph.D., Kimley-Horn & Associates
Contact: 520-977-5753
Doug.gettman@kimley-horn.com

March 2012
Contents
Overview ......................................................................................................................................... 3
Integration of the Kadence Adaptive System with KITS ............................................................... 4
Adaptive Algorithms ....................................................................................................................... 7
  Cycle Tuning ............................................................................................................................... 7
  Offset Tuning .......................................................................................................................... 10
Captured flow ............................................................................................................................ 14
  Distributed Offset Adjustment ............................................................................................... 14
  Including Safety in the Assessment of Offset Performance .................................................. 14
Split Tuning .................................................................................................................................. 16
  Split Constraints ...................................................................................................................... 16
  Calculation of Duration Constraints ...................................................................................... 16
  Estimating Phase Utilization ................................................................................................. 18
  Estimation of Split Utilization ............................................................................................... 20
  Balancing Utilization Levels ................................................................................................. 21
  Incorporating Safety Assessment in the Optimization of Splits ............................................ 22
Phase Sequence Changes .......................................................................................................... 24
  Protected / Permitted Left-Turn Mode Changes (future) ......................................................... 25
Kadence Adaptive Control System Summary .............................................................................. 25
Future Directions .......................................................................................................................... 29
Overview

The Kadence (from the English noun *cadence* meaning balanced, rhythmic flow) adaptive control system was developed as a SBIR in a joint venture with Kimley-Horn & Associates (headquarters in Raleigh, NC with 56 offices nationwide) and Sabra-Wang & Associates (headquarters in Baltimore, MD with several offices in the Baltimore-Washington region). The development and software integration staff was done by the KITS arterial and freeway management systems team. The system optimizes traffic signal timing to balance performance benefits for safety and efficiency. The system is not intended to replace or obviate the need for sound traffic engineering but rather supplement the traffic engineer’s toolbox with another tool that can handle fluctuations in demand and short and long-term changes in land use and traffic patterns.

Kadence is comprised of four principle algorithms for tuning signal splits, offsets, cycle time, and phase sequence. Switching between protected, protected-permitted, and permitted-only left turns (supported by appropriate field infrastructure) is planned for implementation in 2013. A safety performance function is also included that allows the real-time system to predict the changes in traffic conflict rates when adjusting signal timing parameters. First, the system considers efficiency as the primary objective in each optimization algorithm and then checks if the safety performance is also improved. When both safety and efficiency are improved, these new signal timing settings are sent to the field controllers. If efficiency is improved but safety is predicted to be degraded, no new parameters are downloaded to the field. Due to the experimental nature of the safety prediction function, this feature can be disabled to provide the system greater flexibility to make efficiency improvements.

The Kadence system builds upon the operation of the ACSLITE concept developed between 2002-2007 for FHWA by Dr.s Gettman (currently with KHA), Shelby (currently with Econolite), and Head (currently with University of Arizona). In the ACSLITE approach, new signal timing parameters are downloaded to field controllers every 3-4 cycles. The field controller then begins operating in an actuated-coordinated or actuated-free mode with these new settings. Based on past experience with adaptive systems that override the controller’s timings every second (RHODES, OPAC, SCOOT, SCATS, and InSync), this methodology of downloading new timings is more reliable, safer, and less error prone. In addition, the approach of ACSLITE is proven to require minimal capital investment, infrastructure, detectors, configuration, and calibration. The system operation has been validated in over 10 deployments nationwide to produce improvements to travel time and system delay over actuated-coordinated operation with TOD plans.

In addition to the implementation of three optimization algorithms (cycle tuning, cycle selection, and phase sequence selection), several other key enhancements have been implemented. The system requires three cycles of good observations of phase timing and detector data before making decisions but now checks every 30 seconds to identify if that requirement is satisfied. This vastly improves the responsiveness of the system versus the fixed-horizon scheme (5 minutes, 10 minutes) implemented in the baseline ACSLITE system. The offset tuning algorithm has also been enhanced to search offsets in a range instead of only considering fixed changes (+5, 0, -5). By selecting larger search bounds (say, +/-20s), Kadence can quickly find the correct offset solution when the current solution is particularly poor. Kadence has been integrated with VISSIM using Virtual D4 controller firmware. This provides rapid prototyping.
Overview of Kadence Adaptive Control System

and testing of any real-world situation with accurate controller parameters and real-world operation.

Integration of the Kadence Adaptive System with KITS

License Fee and Integration Expenses
The Kadence adaptive control system is integrated as a component of the KHA KITS arterial and freeway management system. This component is available to KITS clients for a one-time license fee, integration costs with the agency’s field controllers, configuration and training. The license fee will allow the agency to use the adaptive control module on any or all of the intersections throughout the system. As with traditional actuated-coordinated systems, we recommend the configuration of groups of signals that are managed with adaptive control as a system. These adaptive groups are notionally one-to-one with “sections” in KITS. A maximum of 32 signals are allowed in each adaptive group. A stand-alone mode is also supported so that the Kadence can be deployed as a “master” controller without the rest of the KITS signal management system software, if desired.

Second-by-Second Polling
The ACSLITE adaptive control approach requires the use of NTCIP or AB3418E National standards communications for phase timing and detector data. This makes the system portable with essentially any field controller from any vendor including ASC/3, McCain 233, SEPAC, LACO4, TSCP, D4, City of LA, and NextPhase. To provide the most flexible support for “any” controller type, Kadence aggregates the MOE data using second-by-second polling of phase timing and detector data from intersections. The original ACSLITE system and approaches used by other vendors require the phase timing and detector data to be aggregated on the local controller which requires a firmware update. Kadence does not require these firmware changes.

In addition, this requirement for second-by-second polling of detector data requires high speed (i.e. IP) and high-reliability communications to any field controllers that are put under adaptive control. Based on the recent successes we have had in converting copper serial multi-drop lines to VDSL (IP), the cost of communications upgrades is relatively low.

Field Detector Requirements
Detection requirements for the adaptive system are as follows:

- Stop-bar detection for all phases that are adaptively controlled. These detectors can be any length. Lane-by-lane detection (i.e. separate lead-in cables from the field loops to the cabinet) is strongly preferred, but not required. Any detection technology (video, loops, magnetometer) is supported. Any phase can be designated to be excluded from optimization, if desired, or if the phase is anticipated to be run in recall because of lack of detection).
- Advance detection for all coordinated phases. These loops or zones should be relatively short (i.e. 6x6 zones or loops in each lane). Lane-by-lane detection (i.e. separate lead-in cables from the field loops to the cabinet) is strongly preferred, but not required. Any detection technology (video, loops, magnetometer, radar) is supported. Exit detection is supported, as is mid-block detection.
Kadence runs on Windows PC platform and requires no additional field hardware in the cabinet. The system can be deployed as part of KITS or as a stand-alone management platform like a “master” controller using a field-hardened PC with solid-state HD and fanless processor and associated temperature –tolerant peripherals.

Figure. Basic Controller Configuration Setup
Figure. Example display for Cycle, split, offset, and sequence tuning performance
Overview of Kadence Adaptive Control System

Adaptive Algorithms

Kadence builds additional functionality on top of the base ACSLITE system. The ACSLITE optimization system was developed by Dr.s Gettman (currently with KHA), Shelby (currently with Econolite), and Head (currently with University of Arizona) for FHWA from 2002-2007. Field trials and deployment of the ACSLITE sites was completed in 2006. All field deployments resulted in improvements to system delay and arterial travel time. ACSLITE was developed to address many of the shortcomings of the ACS algorithms (RHODES, OPAC) developed by FHWA in the mid 1990s. ACSLITE and Kadence are thus based on a simple traffic model that has few tunable parameters and requires minimal calibration. ACSLITE originally consisted of only split and offset tuning methods. Kadence includes additional algorithms for cycle selection, cycle tuning, and phase sequence selection. In all of the descriptions below, we discuss the methodology including the consideration of improved safety performance. This trade-off check can be disabled to increase the system’s ability to maximize efficiency improvements.

Cycle Tuning

Cycle time is selected on a section- or arterial-wide basis to facilitate progression and to provide adequate capacity to operate all of the signals under capacity. Most often, cycle times in adaptive control systems are chosen according to a heuristic rule or, in the case of real-time second-by-second systems, the concept of cycle time is transient – the cycle length can change in each iteration as the system operates more like “free” than in coordination. For example, SCATS uses a heuristic that might be termed the “90% rule”. The SCATS cycle for a section is ratcheted up or down based on keeping the most saturated intersection in the section just at 90% degree of saturation (DoS). This is a reasonable strategy to follow and guarantees that the cycle selected will not artificially cause congestion in certain approaches. In a simplistic fashion, if the cycle time is increased by X seconds, then every phase on the controller gets a proportion of the additional time. For example if 4 seconds are added to the cycle time and there are four phases per ring, one additional second is provided for each phase split.

This strategy will tend towards longer cycles during peak periods as traffic demand builds, which is appropriate, although there is recent research indicating that when the conditions are extremely oversaturated, shorter cycles will provide more efficient throughput. Kadence is being designed to take advantage of the improvements identified by Gettman in NCHRP 03-90, operation of traffic signal systems in oversaturated conditions. This will improve the capability of Kadence to provide sound decisions during incident response conditions, such as heavy diversion of flows from a freeway to a parallel arterial or frontage road system.

The Kadence approach extends from this “critical intersection” algorithm providing a small improvement to this heuristic to consider some k highest saturation levels (or a determination of critical approaches or phases, such as the principal progression phases on an arterial) throughout the network and only increase the cycle time when a minimum number of those critical links are above the specified level of phase utilization. Similarly, decreasing the system cycle when all approaches are less than (say, 60% phase utilization) may not be desirable to maintain adequate progression. To combine the effects of both safety and efficiency, we include the evaluation of our safety performance function in the algorithm for adjusting the cycle time as shown in the following Figure.
In this manner, Kadence considers the possible improvements to both safety and efficiency by first checking the efficiency metric for the cycle time – are all of the $k$ highest saturation levels above or below 90%? If the answer is yes that there are at least $k$ movements with saturation levels above 90%, we consider an increase to the cycle time. If this change results in improvement to the safety metric (i.e. reducing the total predicted conflicts), Kadence continues in this direction of search until one or more of the $k$ highest saturation levels have fallen below 90% saturation (or whatever threshold is determined to be appropriate by the user). As stated earlier, the calculation of the safety effects can be disabled if desired to provide greater flexibility in improving efficiency.
Similarly in the opposite direction of cycle change, we consider reducing the cycle time if there are more than $k'$ (where it could be true that $k$ is not the same as $k'$) movements with saturation level below 70% (or whatever threshold is determined to be appropriate by the user). If this is true, we implement a reduction to the cycle time. The system cycle time is reduced again in the next iteration until there are no longer $k'$ movements with saturation level below 70%.

**Cycle Selection**

In addition to cycle tuning, we have implemented an algorithm which evaluates a quick decision to jump to a different cycle time that is next in the planned TOD schedule. This can be used in conjunction with the cycle tuning method, or as a replacement to the cycle tuning method. This algorithm can allow the system to jump to the next cycle earlier or stay in the current cycle longer than the existing TOD schedule. This allows the system to adapt quickly to changes in a shift to the beginning or end of peak periods. The same evaluation approach for the incremental cycle tuning algorithm described previously is used but rather than changing the cycle just a few seconds, the system enables the new cycle time immediately and recalculates splits and offsets appropriately. This is illustrated in the following Figure.

![Figure. Concept of implementing the next cycle time earlier or later than scheduled](image_url)
Offset Tuning

Cycle time tuning affects all of the intersections in the network if they are all operating in a coordinated mode. Tuning offsets improves progression performance along primary routes for phases that are coordinated. Offset tuning algorithms are particularly straightforward. The proven and robust methodology used in ACSLITE is used in Kadence.

The concept of the data-driven offset adjustment algorithm is summarized in two simple statements: (1) use detectors several hundred feet upstream of the signal to construct cycle-based (“cyclic”) profiles of traffic flow arriving to the intersection, and (2) adjust the offset to maximize the number of vehicles arriving during the green phase. This concept is then expanded to consider and mitigate the effects of such modifications to the offset value for two-way traffic and the effects of changes at this intersection on adjacent intersections. Periodically, small, incremental adjustments are made to the offset to maximize the total proportion of cyclic flow arriving to a green light.

It is assumed that an initial solution (plan data – cycle, splits, and offset) has already been developed and that the original offset may be less than optimal. A user-configurable maximum deviation from the original setting (either an increase or decrease to the offset value) is defined for each offset to restrict the algorithm from drifting too far away from the original solution. The user can also specify that this value is unbounded, which allows the system to search for any offset.

The figure below illustrates the detector locations used for offset tuning. There is one detector station for each coordinated approach. Intersection 1 is referred to as the upstream intersection and intersection 2 is referred to as the downstream intersection. Traffic volume and occupancy is measured at some point between 1 and 2 by a detector in each lane. These detectors can be located where typical advanced detectors are located (200-300ft from the intersection). Placing detectors further upstream can improve the quality of the flow rate measurements, and reduces the possibility of vehicles queuing over the detectors when the light is red. It is also not necessary to have one detector per lane returned to the controller in a separate amplifier, but this practice will improve performance of the algorithm. Exit detection can also be used, if available.
Assuming that the traffic signals at both intersections are using the same cycle length, and that traffic volumes and turning proportions are reasonably steady over time, it is expected that the detector will measure approximately similar recurring patterns of flow each cycle. These patterns of flow are referred to as cyclic flow profiles.

Plots of the flow profile data (volume and occupancy observations) as a function of the local cycle time of the controller (time is on the x-axis; not direction from West to East) as shown in the Figure below. The magnitude of the volume and occupancy is indicated by the height of their corresponding bars in each row of the chart. The height of the bars in each row is scaled by the maximum value observed in that row (equal heights in different rows do not necessarily indicate the same volume or occupancy value). These profiles indicate that it is typical over the last few cycles that traffic is arriving near the beginning of the local cycle time for this approach. Secondary platoons and individual vehicles also show up throughout the cycle, due to turning flow on the cross street phases.
Figure. Example of volume and occupancy data from a typical advance detector

The figure below illustrates an example of phase timing history observed over the last several cycles at an intersection.

Figure. Example of phase timing for each of the last several cycles
The number and color of each column in the timeline corresponds to the active phase interval (green, yellow, and red) displayed by each ring at that time in the cycle. All subsequent cycles shown below the first row are actual data recorded from a field test controller with the most recent data at the top and progressing back in time as you go further down the display.

Each cycle begins at the local zero time, which is labeled on the left. Note that the duration of non-sync, actuated phases (typically phases other than 2 & 6) are variable, and one or more of these phases may be skipped in any given cycle. Thus, the time at which the controller returns to the sync phases (typically phases 2 & 6) can and does vary from cycle to cycle.

The cyclic flow profiles are then averaged to generate a single, representative cyclic flow profile for the flow profile detector. Each link, and thus each flow profile detector on that link, is associated (via user configuration) with a corresponding phase at the downstream intersection. Note that in the case of an arterial, the progression phases generally correspond to the coordinated phases on the controller, but any phase can be designated as the progression phase if the primary progression movement is a turning flow. In a grid network all major through phases (coordinated and non-coordinated) might be configured as progression phases for their corresponding approaches. The figure below shows that during a portion of the cycle, the progression phase is green 100% of the time, starting at local time 50 and ending at local cycle time 0 (or 80). Each cycle one or more non-sync phases typically gap out early, and in such cases, the controller returns to the coordinated phase earlier than is required (and is typically termed “early return to green”). Figure 6 illustrates this early return to green behavior with the tapering percent-green bars prior (to the left of) to the programmed start of main street green split (local time 50 seconds). As shown, this progression phase started as early as local cycle time 27 in at least one cycle during the last ~10 cycles.

![Figure](image_url)

**Figure.** Example of average cyclic volume and occupancy profiles

Note that occupancy, rather than volume, is the preferred detector measurement used to generate flow profiles.

The flow profile scenario shown above is an example of the performance of a good offset for one-directional travel. The arriving platoons are indicated by the cluster of relatively tall occupancy bars between local cycle time 40 and local cycle time 75 which corresponds to the green of the service phase.
Captured flow
The effectiveness of offset settings (or the relative difference between the settings) at upstream and downstream intersections is measured or quantified by calculating the % arrivals on green, or the “captured flow”. This performance measure is a surrogate for stops and delay. Specifically, the captured or progressed flow is the amount of flow (in units of vehicle-seconds of occupancy) arriving to the stop-line at a given point in the cycle multiplied by the percent of time the progression phase is green at that time during the cycle. The algorithm evaluates different offsets by calculating the captured flow on each approach and selecting the offset that maximizes the total amount of captured flow.

Distributed Offset Adjustment
Kadence uses a distributed offset adjustment method. This approach makes offset adjustments for each controller independently but with consideration of the effects of each independent decision on adjacent signals. Each controller considers a range of offset settings: no change, adjust up to \( \Delta \) seconds earlier, or adjust up to \( \Delta \) seconds later. The adjustment maximum step size, \( \Delta \), is a user-configurable value. If the value is set at 10, for example, Kadence will search offsets in each step in the range of \((+10, +9, +8, +7, \ldots, 0, -1, -2, -3, \ldots, -10)\). The adjustment procedure estimates the amount of cyclic traffic flow progressed for inbound and outbound links of the controller for each of the adjustment options and chooses the option that maximizes the total progressed flow. As a reliability measure, there must be at least 3 cycles of vehicle-occupancy data for every flow profile detector at this intersection and adjacent intersections (as well as the corresponding phase timing status information) to execute the selection algorithm. This provides a level of smoothing so that the changes are not frequently toggling between one offset and another. The ACSLITE project field tests determined that this methodology is effective at making adjustments to offsets to improve progression, while mitigating the effects of controller transition by only making small adjustments for each tuning.

[2012 enhancement: if the difference between the evaluated offsets is not greater than small amount of improvement, say, 5%, the controller will remain at the current offset. This will reduce transition events that do not result in significant improvement to performance.]

Including Safety in the Assessment of Offset Performance
The approach to tuning offsets discussed above is augmented as with the other algorithms to consider both efficiency and safety. Similar to the approach for tuning the cycle time, the algorithm will use the safety performance function as the performance calculator. This approach is detailed in the flow chart below. As with other algorithms, the consideration of safety performance can be toggled on or off. In the Phase I research, offsets had the weakest correspondence to improvements or detriments in safety.
As shown above the general approach is to identify the capture efficiency performance of options to move the offset forward or backward. If one or the other improves performance over the “no change” situation (note that it is not common or probable that both moves improve performance), we check the safety impact of the proposed change. If this change is deemed to both improve safety and efficiency, the algorithm will continue exploring this direction of change for additional improvements. If the change degrades safety, the algorithm will not adjust the offset.
Split Tuning

The approach used to tune splits is driven by collecting the same data as used to tune offsets (phase timing information and second-by-second detector occupancy and volume), but collected from detectors at the stop bar of the intersection. The Kadence methodology is also derived from the algorithms developed in the ACSLITE project. In general, this approach to tuning split times is derived from the concepts used by SCATS and SCOOT which is to equalize the degree of saturation on all the phases at the intersection. In shorthand, this approach is termed “equisat”. To be specific, the algorithm in Kadence minimizes the maximum degree of saturation on any phase, rather than driving all of the saturation levels to the same value. This algorithm also allows coordinated phases to have biased splits, so that progression is protected when the saturation level of the coordinated phase was lower than that of side-streets. Without such biasing, the performance is slanted towards providing adequate LOS on side streets, which neglects the throughput issues. The approach works by balancing a measure of degree of saturation that is termed “phase utilization”. The algorithm is described in more detail in the following sections. This section can be skipped for readers with limited time.

Split Constraints

There are constraints on split adjustments which can be defined using a phase-barrier diagram. A typical phase-barrier sequence is illustrated, with barriers explicitly labeled below.

<table>
<thead>
<tr>
<th>b</th>
<th>1</th>
<th>2</th>
<th>A</th>
<th>3</th>
<th>4</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>A</td>
<td>7</td>
<td>8</td>
<td>b</td>
<td></td>
</tr>
</tbody>
</table>

Figure. Ring diagram with barriers (denoted bold vertical lines) explicitly labeled

In discussing the split adjustment algorithms, it is necessary to refer to certain groups of phases. The collection of phases on a particular ring, between two particular barriers is referred to as a ring group. There are four ring groups, (1, 2), (3, 4), (5, 6), and (7, 8).

A barrier group, is the collection of all phases (or all ring groups) between two particular barriers. In the example above, there are two barrier groups (1, 2, 5, 6) and (3, 4, 7, 8). The algorithm uses these groups to swap split time from one phase to another in order to determine the set of split adjustments that satisfy the cycle time.

It is necessary to first determine the legal range of adjustments for each split, before solving for the set of split values recommended for a given controller. This includes consideration of minimum times, maximum times and pedestrian crossing restrictions. These constraints are also important to calculate for each ring and barrier group. While this may seem trivial, it is important to accurately calculate these values before searching the optimization space as this significantly improves the calculation efficiency.

Calculation of Duration Constraints

1. Compute the minimum, current, and (initial) maximum split durations for each phase \( p \), \( p \in P \).
a. Compute $s_p^{\min} = \max\{g_p^{\min}, g_p^{\max_{\text{initial}}}, g_p^{\text{walk}}, g_p^{\text{ped clear}}\} + y_p + r_p$, for each phase $p$. Note that the ASC MIB objects uploaded from each controller are in mixed precision (some in seconds, others in tenths of a second). These values are combined in tenths-of-a-second units, and the rounded up to the nearest second. If phase $p$ is omitted in the current pattern, then $s_p^{\min}$ is set to zero.

b. The value $s_p^{\text{cur}}$ is an ASC MIB object uploaded directly from the controller for each pattern. If phase $p$ is omitted in the current pattern, then $s_p^{\text{cur}}$ is set to zero.

c. Initially set $s_p^{\max} = g_p^{\max} + y_p + r_p$, where $g_p^{\max}$ is set depending on the maximum mode (max1, max2, or max inhibit) currently used by the controller. If max inhibit is the current mode, then $s_p^{\max}$ is set to 255 seconds. If phase $p$ is omitted in the current pattern, then $s_p^{\max}$ is set to zero. Note that $g_p^{\max}$ is specified in seconds, whereas yellow and red intervals can be specified in tenths-of-a-second. These values are combined in tenths-of-a-second precision and then rounded down to the nearest second such that $s_p^{\max}$ is in seconds.

d. Ensure the current split is within the configured minimum and maximum constraints. If it is not true that $s_p^{\min} \leq s_p^{\text{cur}} \leq s_p^{\max}$, then STOP; the configuration data is invalid.

e. If the adjustment process is constrained such that only incremental adjustments from the current value are allowed, or a maximum cumulative deviation from an underlying “baseline” split value, then adjust the minimum and maximum constraints here. If the constraint on the cumulative deviation from the baseline split is not satisfied by the current split value, then STOP; this constraint is not currently achievable.

f. If there are no detectors associated with the current phase, then STOP; this is a configuration error.

2. For each group, compute the minimum, current, and (initial) maximum durations for each group $g$.

a. For each ring $r$, compute the minimum, current, and (initial) maximum ring-group duration for the ring-group in barrier group $g$ on ring $r$ as follows:

$$d_{rg}^{\min} = \sum_{p \in P_{rg}} s_p^{\min}, \quad d_{rg}^{\text{cur}} = \sum_{p \in P_{rg}} s_p^{\text{cur}}, \quad \text{and} \quad d_{rg}^{\max} = \sum_{p \in P_{rg}} s_p^{\max}.$$ 

b. Now compute the barrier group durations as follows:

$$d_g^{\min} = \max_{r \in R} \{d_{rg}^{\min}\},$$

$$d_g^{\text{cur}} = \max_{r \in R} \{d_{rg}^{\text{cur}}\},$$

$$d_g^{\max} = \min_{r \in R, d_{rg}^{\min} > 0} \{d_{rg}^{\max}\}.$$ 

In general, there will always be at least one phase in each barrier group, hence there will be one ring $r$ such that $d_{rg}^{\max} > 0$; but if not, then set $d_g^{\max} = 0$.

3. Calculate the revised maximum duration for each barrier group, its corresponding ring-groups, and the maximum phase splits of their corresponding phases.

a. Calculate the revised maximum duration for group $g$, under the assumption that each other group must time at least its minimum duration, as follows:

$$d_g^{\max} = C - \sum_{g' \in [G: g' \neq g]} d_{g'}^{\min}.$$
b. Revise the maximum duration for each ring-group on each ring \( r \) of the barrier group \( g \), such that it is not greater than the maximum barrier group duration of group \( g \), as follows: 
\[
\max_{rg} d_{rg} = \min \{ \max_g d_g, \max_{rg} d_{rg} \}.
\]

c. Ensure that the current ring-group duration (i.e., sum of splits) is not greater than the maximum ring-group duration possible with the cycle. If \( d_{rg}^{\text{cur}} > \max_{rg} d_{rg} \), then STOP, there is a configuration problem.

d. Calculate the revised maximum splits for each phase \( p \) in the ring-groups from each ring \( r \) of the barrier group \( g \), under the assumption that the maximum duration of the ring-group may not be exceeded and each other phase in the ring-group must time at least its minimum duration. This is calculated as follows:
\[
s_{rg}^{\text{max}} = d_{rg}^{\text{max}} - \sum_{p' \in \{r_x \cup p \}} s_{p'}^{\min}.
\]
Ensure that the current split is not greater than the revised maximum split. If \( s_{p}^{\text{cur}} > s_{p}^{\text{max}} \), then STOP, there is a configuration problem.

Once the minimum and maximum phase split duration constraints have been determined using the preceding procedure, the multi-ring split adjustment algorithm may commence.

**Estimating Phase Utilization**

The split adjustment algorithm is based on the notion of balancing the utilization of all phases of a signal controller. Prior to discussing the algorithm itself, it is necessary to define:

- What phase utilization is,
- How it is measured, and
- How it is estimated for split durations other than what is currently in use by the controller.

*Phase utilization* is the effectively utilized percentage of available split time. It is analogous to the *degree of saturation* concept, which is also referred to as the volume-to-capacity or V/C ratio. Utilization is used instead of degree of saturation since the degree of saturation is calculated using volume and capacity counts or rates. Utilization is calculated using ratios of used green time to available green time. The used green time comes from the occupancy of the detector.

The figure below illustrates a typical detector layout for measuring phase utilization with a detector placed at the stop bar for each lane. Each detector is associated with the phase that serves traffic flowing through its corresponding lane. Detector dimensions do not have to be the theoretically “best possible length”.
The methodology measures the demand for a phase by monitoring the occupancy of the phase during green. Demand is measured in terms of time, rather than volume. Utilization is given by the ratio of demand time to available green time. This is illustrated in below.

Figure. Measuring phase utilization for coordinated-actuated controllers

Kadence polls controllers for phase timing and detector data and aggregates the data over time (i.e. combines several consecutive once-a-minute poll responses) to construct estimates of occupancy during green, green phase duration, and utilization.

The example above shows where ten seconds of green is served, seven seconds of which the detector is occupied. For a fixed-time controller, this corresponds to 70% utilization. However, in the context of coordinated, actuated controller the capacity of the phase is measured as the amount of available green time. In this case, the phase started timing green 2 seconds early due to a prior phase gapping out early. In this case, it could serve up to 12 seconds of green until it is
forced-off, but it gaps out after 10 seconds of green. The notion of available green includes the remaining time to the force-off or maximum green, whichever comes first. It is also important to consider when phases have been skipped, due to lack of demand.

Similarly to the other adjustment algorithms, the splits will not be tuned without at least collection of at least three observations of each phase to be tuned, while the controller is in coordination (i.e. when the controller is in transition or preemption, the algorithm will not use this data in tuning splits).

[2012 note: data during transition is being evaluated to be used for split adjustments to improve the responsiveness of Kadence]

Having satisfied this condition, the algorithm then calculates the utilization of alternate split durations for each phase using the following variables and equations, where:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_p$</td>
<td>Number of observations of phase $p$</td>
</tr>
<tr>
<td>$o_{ip}$</td>
<td>Occupancy of phase $p$ green time during observation $i$ (0-100%)</td>
</tr>
<tr>
<td>$g_{ip}$</td>
<td>Green time served by phase $p$ during observation $i$ (seconds)</td>
</tr>
<tr>
<td>$a_{ip}$</td>
<td>Available green time for phase $p$ during observation $i$ (seconds)</td>
</tr>
<tr>
<td>$d_p$</td>
<td>Average demand (seconds occupied green) per cycle for phase $p$</td>
</tr>
<tr>
<td>$a_p$</td>
<td>Average available green time (seconds) per cycle for phase $p$</td>
</tr>
<tr>
<td>$c_p$</td>
<td>Clearance time (seconds of yellow and all-red) associated with phase $p$</td>
</tr>
<tr>
<td>$u_{pt}$</td>
<td>Utilization estimate for phase $p$ with split of duration $t$ seconds</td>
</tr>
</tbody>
</table>

### Estimation of Split Utilization

1. If $N_p < 3$, then STOP. There is not enough data for split adjustment.

2. Compute average demand, $d_p = \frac{\sum_{i=1}^{N_p} o_{ip} g_{ip}}{N_p}$.

3. Compute average available green time, $a_p = \frac{\sum_{i=1}^{N_p} a_{ip}}{N_p}$.

4. For each feasible split duration $t$, between $s_p^{\min}$ and $s_p^{\max}$, estimate the utilization of that split assuming current demand remains the same and the phase starts on average at the same point in the cycle.
   a. If $a_p + t - s_p^{\text{cur}} > 0$ (i.e. the available green time of such a split duration is non-zero), then set $u_{pt} = \frac{d_p}{a_p + t - s_p^{\text{cur}}}$.
b. Otherwise if \( a_p + t - s^\text{cur} \leq 0 \) (i.e. there is no available green time), then either set \( u_{pt} = 0 \) if \( \bar{d}_p = 0 \) (i.e. there is no demand), or set \( u_{pt} = \infty \) if \( \bar{d}_p > 0 \).

After calculating these estimated utilization levels for all alternate split durations, an iterative algorithm is executed to select the combination of split values that satisfies the constraints on each phase and phase group (i.e. ring and barrier groups) and minimizes the maximum utilization of any phase on the controller. This procedure is discussed in the next section.

**Balancing Utilization Levels**

Each phase is assigned a *utilization* measure that approximates the degree of saturation of that phase, which ranges from 0 to 100% (or higher if oversaturated). Utilization is estimated for each phase, for the full range of possible split durations, as discussed in preceding sections. The figure below is an example of utilization estimates for a dual-ring, eight-phase controller, where the utilization of phase 3 is very high. A chart of the estimated utilization of phases after the algorithm has adjusted the splits to minimize the maximum utilization of any phase on the controller. The changes in split times are listed in the Table. The red circle highlights the change of the most utilized phase from 100% to 72%. Also note that the new maximum phase utilization is now on phase 6, which has increased from 72% utilization to an estimated 76% utilization. Note that the result of the adjustment is not the exact same utilization level on each phase.

![Figure. Utilization of phases before and after split adjustment](image_url)
Table 1. Example utilization of phases before and after split adjustment

<table>
<thead>
<tr>
<th>Split Times (sec)</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
<th>Phase 5</th>
<th>Phase 6</th>
<th>Phase 7</th>
<th>Phase 8</th>
<th>Utilization (%)</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
<th>Phase 5</th>
<th>Phase 6</th>
<th>Phase 7</th>
<th>Phase 8</th>
<th>Maximum Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Adjustment</td>
<td>14</td>
<td>21</td>
<td>13</td>
<td>22</td>
<td>10</td>
<td>25</td>
<td>15</td>
<td>20</td>
<td>36%</td>
<td>61%</td>
<td>100%</td>
<td>72%</td>
<td>35%</td>
<td>72%</td>
<td>40%</td>
<td>34%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Post-Adjustment</td>
<td>11</td>
<td>21</td>
<td>16</td>
<td>22</td>
<td>10</td>
<td>22</td>
<td>17</td>
<td>21</td>
<td>53%</td>
<td>61%</td>
<td>72%</td>
<td>72%</td>
<td>35%</td>
<td>76%</td>
<td>33%</td>
<td>32%</td>
<td>76%</td>
<td></td>
</tr>
</tbody>
</table>

As indicated previously, the primary objective of the split adjustment algorithm is to minimize the maximum degree of utilization across all phases. These objectives achieve the general effect of balancing the degree of utilization across phases and will have the effect of minimizing delay.

**Incorporating Safety Assessment in the Optimization of Splits**

Similar to the approach used for cycle time and offset tuning, the algorithm incorporates evaluation of safety into the optimization by utilizing a safety performance function. This process is illustrated in the flow chart below. The process is similar to the processes developed for offsets and cycle time.
Figure. Flow chart of the split optimization process including safety analysis
Phase Sequence Changes

Phase sequence affects both progression and delay at an intersection with respect to measures of efficiency. In Phase I of the research that led to Kadence, we also found that the sequence can also affect safety of the intersection, particularly when an intersection operates in coordination in a signal system. In our parameter tuning approach, improvements to both safety and efficiency can be made by analyzing the phase utilization measures for each signal phase and the total capture efficiency of the coordinated phases. To illustrate the concept of the algorithm, we only consider one barrier group (e.g. phases 1, 2, 5, 6 in a standard dual-ring quad intersection). The equivalent rules apply to phases (3, 4, 7, 8) in the other barrier group. In lieu of a flow chart we present the concept as a table of decision rules. In the left column, the current operating phase sequence is listed, lead-lead, lead-lag, lag-lead, and lag-lag. The top row lists the potential phase sequence that we will evaluate changing to. The cells of the table indicate the rules that would justify a change from one sequence to another. For example, a change from lead-lead to lag-lead would be predicated if:

- Phase 5 has a heavier utilization than phase 1
- Phases 2 and 5 have heavier total utilization than phases 1 and 6

In this example case, phase 1 would change to a lagging phase, which moves the offset to coincide with the end of phase 2 (most controllers reference the offset to the yellow time of the first coordinated phase) instead of the end of both phases 2 and 6. At this intersection, the offset value does not have to change. However the change to the time that phase 2 will occur during the cycle will change the time that the traffic arrives at the intersection downstream from phase 2 and also the capture efficiency of the traffic that arrives to phase 6 from the intersection upstream of that phase. For example, if phase 2 services northbound traffic and phase 6 services southbound traffic, the intersection to the North will experience traffic arriving earlier in its cycle and the intersection to the South will have the traffic arrive later in its cycle. Offsets are adjusted after the sequence change is evaluated.

| Current sequence \ potential next sequence | Lead-lead (12|56) | Lead-lag (12|65) | Lag-lead (21|56) | Lag-lag (21|65) |
|-------------------------------------------|-----------------|-----------------|-----------------|-----------------|
| Lead-lead (12|56)                      | N/A             | PU5 < light     | PU1 < light     | PU1 < light     |
|                                           | Diagonal(16)    | Back diagonal(25) dominates diagonal(16) | Back diagonal(25) dominates diagonal(16) | Back diagonal(25) dominates diagonal(16) |
| Lead-lag (12|65)                      | PU5 > heavy     | N/A             | PU1 < light     | PU1 < light     |
|                                           | Diagonal(15)    | PU5 > heavy     | Back diagonal(25) dominates diagonal(16) | Back diagonal(25) dominates diagonal(16) |
|                                           | dominates back diagonal(26) | Lag(25) dominates lead(16) | Lag(25) dominates lead(16) | Lag(25) dominates lead(16) |
| Lag-lead (21|56)                      | PU1 > heavy     | PU1 > heavy     | N/A             | PU5 < light     |
|                                           | Back diagonal(15) dominates diagonal(26) | Lag(16) dominates lead(25) | Back diagonal(26) dominates diagonal(15) | Back diagonal(26) dominates diagonal(15) |
| Lag-lead (21|65)                      | PU1 > heavy     | PU1 > heavy     | PU5 > light     | N/A             |
|                                           | PU5 > heavy     | PU5 > light     | Back diagonal(25) dominates back diagonal(16) | Back diagonal(25) dominates back diagonal(16) |
|                                           | Lag(15) dominates lead(26) | Lag(16) dominates lead(25) | Lag(25) dominates back diagonal(16) | Lag(25) dominates back diagonal(16) |

Similar to the evaluation algorithms for cycle, splits, and offsets, after evaluating the efficiency impact of a potential change to the phase sequence, we next check the effect on the safety by
evaluating the regression equation. If the safety is improved, the change is made, otherwise if the safety is degraded, we do not consider the phase sequence change. The evaluation of safety can be enabled or disabled.

**Protected / Permitted Left-Turn Mode Changes (future)**

The research in Phase I showed that the mode of left-turn operations has a significant effect on the safety of an intersection with permitted left turns creating the most conflicts and, not surprisingly, protected left-turns creating the least number of conflicts. Crossing crashes are among the most severe crashes that occur. Efficiency is affected in a slightly different order with permitted being the least efficient, protected being next, and protected-permitted having the highest level of service for the same service volume (assuming the service volume being high enough to require protected-permitted operation). Similarly to the algorithm for selecting phase sequence, we present the algorithm for modifying left-turn accommodation by use of a table. The left-most column lists the current left-turn treatment for a given left-turn. The top row lists the left turn treatment being considered. The cells of the table list the conditions that would justify change from one left turn treatment to the other.

<table>
<thead>
<tr>
<th>Current LT treatment / next left-turn treatment</th>
<th>Permitted only</th>
<th>Protected</th>
<th>Protected-Permitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permitted only</td>
<td>N/A</td>
<td>Utilization of left turn lane is heavy</td>
<td>N/A</td>
</tr>
<tr>
<td>Protected</td>
<td>Utilization of left turn phase is very low</td>
<td>N/A</td>
<td>Utilization of left-turn phase is very high Opposing utilization is low to moderate</td>
</tr>
<tr>
<td>Protected-Permitted</td>
<td>N/A</td>
<td>Utilization of left turn protected portion is moderate to low</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Similarly to the other algorithms, we evaluate the efficiency measure first and then compare any potential detrimental effects on safety by evaluating the trade-off value. If the trade-off value of changes detrimental to safety is acceptable, the change to left-turn operational mode is made. Otherwise the current operating mode is retained. This algorithm has been postponed for implementation at a later time due to the complexities of recommending changes to this type of operation, and the need for a standard way (or a suite of customized methods) to communicate these changes to controllers, particularly with the emerging standard of flashing yellow arrow versus “doghouse” 5-section heads or other alternative displays used by many agencies (flashing red arrow, etc.).

**Kadence Adaptive Control System Summary**

The Kadence adaptive control system was developed as a SBIR in a joint venture with Kimley-Horn & Associates (headquarters in Raleigh, NC with 56 offices nationwide and Sabra-Wang & Associates (headquarters in Baltimore, MD with several offices in the Baltimore-Washington
The principal development and software integration staff reside in the Phoenix, AZ office of KHA as part of the KITS arterial and freeway management systems team. The system focuses on the development of real-time signal timing methodologies and algorithms that balance or optimize both safety and efficiency. The system is not intended to replace or obviate the need for sound traffic engineering but rather supplement the traffic engineer’s toolbox with another tool that can handle fluctuations in demand and short and long-term changes in land use and traffic patterns.

Five principle algorithms were developed for tuning splits, offsets, cycle time, and phase sequence. Switching between protected, protected-permitted, and permitted-only left turns will be added in 2013. A safety performance function was developed that allows the real-time system to predict the changes in traffic conflict rates when adjusting these signal timing parameters. First, the system considers efficiency as the primary objective in each optimization algorithm and then checks if the safety performance is also improved. When both safety and efficiency are improved, these new signal timing settings are sent to the field controllers. Due to the experimental nature of the safety prediction function, this feature can be disabled to provide the system greater flexibility to make efficiency improvements.

There are four principle measures of performance used in Kadence:

- **Phase utilization.** Phase utilization is a surrogate measure of efficiency that represents the degree of saturation of a traffic phase. This measure can be derived directly from the occupancy data measured at stop bar detectors. This measure is used for cycle tuning, cycle selection, split tuning, and phase sequence.

- **% arrivals on green.** % arrivals on green is a measure of efficiency that represents the progression performance of coordinated phases. % arrivals on green can be derived directly from the data measured at upstream detectors on the coordinated phases. This measure is used for offset tuning [future: phase sequence selection].

- **Estimated traffic conflict rate.** Total estimated conflicts per hour is a surrogate measure that represents the estimated effect of changing a traffic control parameter on the intersection safety. This measure is a regression model using a feed-forward neural-network that is trained to learn the relationships between signal timing settings and the crossing, lane-changing, and rear-end conflict rates. [future: weighted conflict rates by type to reflect the anticipated higher severity of crossing and lane-change conflicts that result in crashes]

Each of these measures is used in the adaptive control algorithms as detailed in the previous sections. The five optimization stages are executed independently, but in sequence and with the feedback steps as shown below. First, (Step 1) the split re-allocation algorithm is executed for each intersection in the system. This identifies if any slack green time can be shifted from one or more phases to another, within the current cycle time, to minimize the maximum phase utilization at the intersection. Safety is evaluated by checking that the reallocation either provides a safety benefit by reducing total conflicts or that the reallocation does not exceed a prescribed threshold.
After this re-allocation, the offset adjustment algorithm is executed (Step 2) to identify any modifications to the offsets to improve progression. The total capture efficiency of the offset to “capture” occupancy on the coordinated phases at the subject intersection and its neighbors is calculated to represent the efficiency of the proposed change. Similarly to the split calculation, the safety is evaluated by checking that the new offset either provides a safety benefit by reducing total conflicts or that the proposed change does not exceed a prescribed threshold for the safety/efficiency trade off value.

After the splits and offsets are calculated, modifications to the phase sequence (Step 3) are evaluated with the new split values calculated in Step 1. If any phase sequence modifications are identified that adjust the offset (the departure platoons to adjacent intersections), the offset calculation is re-executed to determine if this change is of further benefit and can be retained, or if the change is detrimental to performance (efficiency or safety).
[Future] Next, Kadence evaluates the potential changes to the protected/permitted settings for left-turns (Step 4), using the newly selected phase sequence, if changed in the previous step. If any left-turn settings are deemed to be beneficial for both efficiency and safety, or beneficial for efficiency and within the safety/efficiency trade off value, the split re-allocation algorithm may
have to be re-calculated, particularly if the left-turn mode is changed from protected to permitted-only. This change in effect omits the left-turn phase setting its split to 0 which frees up additional time in the cycle for other phases. It may not be the best policy to simply provide all of that split for its corresponding companion through phase (e.g. phase 6 if the left turn is phase 1). In turn, if the splits are re-allocated at this step, the offsets, phase sequence, and protected/permitted settings are re-evaluated as well.

Finally, the cycle time adjustment algorithm is evaluated (Step 5). Since cycle time affects all of the intersections in the system, it is important that this adjustment is calculated last, after all of the adjustments/improvements to the individual locations are calculated. As with the phase sequence and protected-permitted settings, if it is deemed beneficial to modify the cycle time, we must re-evaluate the other algorithms within the new value for the cycle (either higher or lower).

**Future Directions**

In addition to the features described above, the following features will be pursued in future projects to enhance the operations further.

1. **“Groundhog Day” elimination.** The current system tunes offsets and splits starting from a pre-configured timing plan. If these settings are inappropriate and the traffic flows are repeatable by day and time, Kadence will re-learn to adjust the settings each time it runs the same pattern. This effect is similar to the Bill Murray film “Groundhog Day” where he is forced to re-live the same day and events over and over. In 2013, the system will be upgraded to calculate a new “starting point” for each TOD signal plan, as well as the most appropriate schedule start point for each timing plan.

2. **Handling of oversaturated conditions.** The current algorithms principles are all based on assumptions of under-saturated operation. While the system can be effective in reducing the duration of oversaturation, or delaying the start of oversaturated operation, in certain complex situations, the decisions made by Kadence may be counterproductive. This is true of every adaptive control system that is available today. Principles from NCHRP 03-90 research developed by Dr. Gettman will be incorporated into Kadence to handle the special situations where downstream congestion creates upstream oversaturation or when phases are starved based on conflicting queues. These features require measurement of the queue length using the upstream detectors and estimation of the oversaturation levels. Gating links and other features related to the research in 03-90 will be implemented.

3. **Suggesting TOD signal timing plans and schedules.** The same algorithms that can be applied in real-time can also be applied to longer periods of time (entire pattern durations) to determine suggested changes to baseline timing settings. This feature can provide more conservative agencies with benefits if they do not wish to adjust timings in real-time but rather want the system to suggest changes but not implement them until they can be reviewed by the engineering staff.