DEVELOPMENT OF A SPATIAL DATABASE FOR INTERSECTIONS IN KENTUCKY

Authors

Eric R. Green, P.E.
Corresponding Author
Research Engineer
eric.green@uky.edu
phone: (859) 257-2680

Chris L. Blackden, Ph.D.
Research Analyst
clblac2@uky.edu
phone: (360) 359-5119

Michael A. Fields
Research Analyst
tony.fields@uky.edu
phone: (859) 257-7562

Mailing Address
Kentucky Transportation Center
College of Engineering
176 Raymond Building
University of Kentucky
Lexington, Kentucky 40506-0281

fax: (859) 257-1815

Date of Submission: 7/30/2015

Number of Words in Text: 6,544

Number of Tables: 2

Number of Figures: 1
Table of Contents

Contents

Abstract ................................................................................................. 3
1. Introduction.......................................................................................... 4
1.1 Background and Objectives ............................................................. 4
1.2 Literature Review ............................................................................ 4
1.2.1 Classifying Intersections ......................................................... 4
1.2.2 Safety Analysis ......................................................................... 4
1.2.3 Development of Safety Performance Functions ....................... 5
2. Methodology ...................................................................................... 5
2.2 Tabular Processing ........................................................................... 6
2.3 Intersection Database ..................................................................... 7
2.3.1 Building Intersections in ArcMap ............................................ 7
2.3.2 Derived Fields and Psuedo-Intersections ................................ 7
2.3.3 Route Ranking System and Master Nodes ............................... 9
2.3.4 Additional Data ......................................................................... 9
2.3.5 Approach Counts ....................................................................... 11
2.3.6 Calculating Entering Traffic ..................................................... 11
3. Intersection Database and Safety Performance Functions ............. 12
3.1 Intersection Classification ............................................................ 12
3.2 Safety Performance Functions ....................................................... 13
4. Conclusions ...................................................................................... 16

References ............................................................................................ 17

List of Tables

Table 1: Intersection Classification ....................................................... 13
Table 2: SPF Parameters ....................................................................... 15
Deciding which intersections in the state of Kentucky warrant safety improvements requires having a comprehensive inventory that contains information on every intersection that is part of the public roadway network. The Kentucky Transportation Cabinet (KYTC) had previously undertaken this task for only those intersections where state-maintained roadways meet. However, this inventory neither accounted for intersections between state- and locally-maintained routes nor was it designed to accommodate regular updates. As such, the Kentucky Transportation Center (KTC) at the University of Kentucky developed a methodology to create and maintain a full inventory of every intersection in the state. This database contains precise location information as well as several safety and operational attributes for each point of an intersection. Each intersection was categorized by several factors to replicate the topology used in the Highway Safety Manual (HSM). Safety Performance Functions (SPF) were developed for each intersection type. These were used to rank each intersection. It is anticipated that this project’s deliverables will be used to increase KYTC’s ability to effectively allocate funds to maintain and improve intersection safety.
1. INTRODUCTION

1.1 Background and Objectives

Intersctions are one of 10 areas of emphasis listed in Kentucky’s most recent Strategic Highway Safety Plan (SHSP). Crashes that occur at intersections represent approximately 25% of all collisions in Kentucky (Kentucky Traffic Collision Facts). Representatives from the Kentucky Transportation Cabinet (KYTC) have expressed that a maintainable intersection database is needed. Because state transportation agencies have become increasingly reliant on spatially explicit data, a spatially-enabled intersection database would benefit many areas of transportation. In 2003, the Kentucky Transportation Center (KTC) conducted a research study on vehicle crashes at intersections. The dataset produced during that research has since become the primary database used by the KYTC for safety analysis and prioritization ranking. Unfortunately, this database was created to analyze safety over a fixed time period and is now out-of-date.

The objective of this research is to develop a comprehensive database that inventories all intersections in the state Kentucky. One that includes the points at which all state-maintained as well locally maintained roadways intersect. The major impetus for creating this database is to equip KYTC with the knowledge it needs to prioritize intersections for safety improvements. To that end, it is important to categorize intersections so they can be compared with one another. Comparisons must focus on similarly functioning sites instead of generating one priority list that ignores important geometric and operational differences. The final database is fully updatable using an automated process. It can identify any changes to roadway geometry that would alter the attributes of intersections. In addition to point features used for locating intersections, this database accounts for each intersection’s zone of influence. This encompasses the area around intersection points at which crashes are assumed to be attributable to the intersection.

1.2 Literature Review

1.2.1 Classifying Intersections

Various methods have been developed to classify intersections and evaluate their safety. Simandl et al. (2015) used Google Maps/ Street View to collect visual observations of an intersection with a preset form or web portal. Observations were linked to an intersection node in spatial database, which were plotted using linear referencing. They limited their study to a small subset of intersections (non-signalized, state-route) in the state of Alabama. Campbell and Knapp (2005) discussed a more comprehensive intersection classification system in the state of Wisconsin. They used parameters such a traffic volume, number of legs, and presence of a median. These categories were then cross-referenced with crash data. Garber et al. (2011) expanded on earlier work by creating Safety Performance Functions (SPF) on the intersection classification system they developed for Virginia (although with far fewer categories than Campbell and Knapp used).

1.2.2 Safety Analysis

In recent analyses, intersections have been ranked using a procedure called critical rate analysis (Green, 2004). Critical rate analysis, however, is problematic for several reasons. First, it assumes a linear relationship between the number of crashes and traffic volume. Recent studies have determined this this is not a valid assumption. For example, sites with very low AADT that have experienced even one crash tend have disproportionately high critical rates. For most roadway and intersection types, the relationship between the number of crashes and traffic volume is exponential (Srinivasan, 2011). The Highway Safety Manual (HSM) outlines procedures to develop SPFs that more realistically represent the relationship between crashes and traffic volume. Unlike critical rate analysis, the SPFs also account for regression to the mean by using an Empirical Bayes statistical method.
1.2.3 Development of Safety Performance Functions

The HSM recommends processes and guidelines for decision making based on safety performance. To properly apply the HSM procedures, SPF’s should be developed based on state-specific crash data. It is anticipated that developing SPF’s using data from local agencies will increase the accuracy and reliability of crash estimates. These estimates can then be used to evaluate the potential effectiveness of alternatives and countermeasures. The HSM’s regression models estimate the predicted average crash frequency for a site based on data from a number of similar sites. In general, SPF’s were developed for each site using historical data. They were then adjusted using the Empirical Bayes (EB) approach to improve the accuracy of estimates and address possible changes from regression to the mean.

2. METHODOLOGY

The spatial intersection database was created using ArcMap and Excel. The final product includes a geodatabase with a polygon feature class called ‘Intersections’ and a corresponding table containing classificatory attributes, including crash statistics. Selected intersections from this table – those satisfying certain data requirements explained below – were then used as inputs for the Safety Performance Function described in Chapter 3.

2.1 Preprocessing

Two files provided by the Kentucky Transportation Cabinet (KYTC) were used as the main inputs for this project – All_Rds_M.shp and an excel file called ‘Kentucky Node Usage.’ All_Rds_M is a measured polyline shapefile representing the Kentucky Road Network. Unlike an ordinary line feature, a measured line feature has a directional distance measure embedded in it, which can be used to plot milepoints from a table via ArcMap’s Linear Referencing Tool. Each record in the node usage table represents a point that can be located along the lines in All_Rds_M, based on a field with a route designation and another field pinpointing the milepoint along this route. Two other fields in the node usage table play a significant role in the intersection-building process. One designates whether the record is a start (S) or end (E) point, the other assigns each point a node ID (NID) number.

The node usage table describes a point wherever a route begins, ends, encounters a gap (called a distant break), or intersects another route. Multiple points can share the same coordinate or ‘stack.’ Single points occur only at dead ends, double points occur when a route changes designation outside of an intersection, and three or more points in a stack indicates the presence of an intersection. For clarity, multiple spatially coincident node usage points are referred to as ‘node stacks’ in this paper. Each point in a node stack is assigned the same NID. A point is defined as S or E depending on the direction in which route milepoints are increasing, except in the case of non-cardinal routes (where the direction is reversed). If a route passing through an intersection (not starting or ending there), it leaves a pair of points with the same milepoint, route designation, and NID but opposite S/E values. The number of points in a node stack generally corresponds to the number of approaches in an intersection, with two major exceptions, one for distance breaks and the other for multi-stack intersections.

Most routes form discrete and continuous lines, as represented in All_Rds_M, but in a significant minority of cases the route is split in one or more places by a distance break. The measurement of distance along the route ignores this gap, so that the two corresponding ends of the split segment are treated as having identical milepoints. This can create problems when plotting points in ArcMap. If a milepoint is located at a distant breaks (which often occur at intersections), the linear referencing tools will arbitrarily assign the point to one side of the gap, often not the correct side. A secondary procedure was developed to correct for these ‘wandering nodes,’ the details of which are not pertinent to the main problems addressed in this paper. Once all nodes were correctly plotted, points representing distant breaks could be ignored since they did not describe a distinct intersection approach.

If we consider each route line to be divided wherever it encounters a node stack, then each node point (once those for distance breaks are discarded) describes one of the endpoints of each segment.

Where a node stack contains three or more of these endpoints, the segments constitute approaches or legs
of a ‘centerline’ intersection. The difference between an intersection and a centerline intersection bears some clarification. The polyline geometry used here describes each route as a two-dimensional line with no width following the center of the road. If the approaches entering an intersection are slightly off-center to one another, it may result in two of more ‘centerline intersections’ in what is functionally one intersection. More complicated arrangements like Y-intersections and crossovers also result in multi-stack intersections or intersections constituted by the combination of more than one centerline intersection. Most of the Kentucky intersections defined in this project included only one centerline intersections and therefore the number of approaches equaled the number of endpoints in that node stack. Counting approaches in multi-stack intersections was more complicated and will described later in the paper. The next section explains the logic used to determine whether or not to merge nearby node stacks into the same intersection.

2.2 Tabular Processing

The original node usage table describes segments as well as points. Each ‘E’ node had a corresponding ‘S’ node upstream of it with the same route ID. Except for non-cardinal routes, the sequence ‘S’-to-'E’ follows the direction of increasing route milepoints in the state of Kentucky (milepoint information was embedded in ‘AllRds_M,’ which is why it was necessary to use this particular file for linear referencing). For the purpose of defining intersections, a portion of the internode line segments was defined as being within the zone of influence for an intersection.

Each node had a milepoint field. This was previously used to plot their points with ‘AllRds_M.’ To create line segments for approaches, each node record was attached to another point a specified distance (100 feet by default) from it along an approach. This segment’s direction was defined by whether or not the node was an ‘S’ or an ‘E.’ For example, a four-legged intersection with no intersections nearby plotted as four 100-foot approach segments radiating out from a central point, which resembled an ‘X’ or plus sign. Where centerline intersections were close enough that their zones of influence overlapped, adjustments were made to the 100-foot distance. Adjustments were also made if the section number of the route indicated that the centerline intersection was part of a larger intersection system (such as an interchange or a Y-intersection). These adjustments took one of two forms. Two or more nodes along the same route were either merged into one intersection or their zones of influence were truncated to avoid overlap.

Microsoft Excel was selected to execute this process in because it facilitated easy rearrangement of the data in the necessary sequence. Excel’s cell reference system also expedited comparisons between adjacent records.

To execute the process, the table was strictly ordered by route ID, S/E identifier, and ascending milepoints (the S/E identifier first had to be reversed for non-cardinal routes). Formulas were developed to compare each node record to the one before or after it (depending on whether the node was ‘S’ or ‘E’) to determine whether the intermediate approach segments should be merged, truncated, or left at default. First it was determined if two consecutive nodes were on the same route. Nodes along the same route that were merged were treated as the same intersection. If two nodes’ zones of influence overlapped, but were not close enough to merge, their approach segment was truncated so that it equaled half the length of the gap minus 1/1000th of a mile (a small gap was left between segments to prevent their touching). The decision to merge or to truncate was based on the type of route, the type of route of the neighboring node, and the length of the gap separating them (i.e. length along the route, not the absolute distance between the two nodes). Any two node stacks with a gap less than 35 feet were merged into the same intersection if the gap was more than 200 feet they were neither merged nor truncated. For gaps 35–200 feet long, nodes were defined as ‘anti-merge,’ ‘merge-happy,’ or neutral. Nodes belonging to routes with the section number ‘000’ were classified as anti-merge and were never merged with one another at distances over 35 feet. ‘Merge happy’ nodes were those that belonged to Y-intersections, non-cardinal routes, or had a neighbor that did. These were merged for distances under 200 feet. Anything else was considered neutral and was merged up to 100 feet, but they were truncated if the distance was between 100 and 200 feet.
In spatial terms, the product of the tabular calculations was the description of a line segment that either started or ended at each node. This segment could then be plotted with the linear referencing tool. For each node record, two fields were created to specify the milepoints at each end of the segment – one identical to the node’s original milepoint, one at a distance (direction determined by the S/E identifier) that was ascertained through the merge/truncate calculations described above. When a merge was performed the second milepoint was identical to that of the next node along the route.

2.3 Intersection Database

2.3.1 Building Intersections in ArcMap
Plotting the modified table with ‘AllRds_M’ using the Linear Referencing tool produced a large batch of line segments. When overlaid atop the previously plotted point features, these radiated outward from the point location of each node stack. In some cases this process linked nearby node stacks together. These segments were used to create polygons that represented each intersection. Rectangular polygons based on each line with a buffer distance of five feet were created. The option for ‘flat-ended’ buffers was applied to preserve the gaps between proximate intersections. The dissolve tool was then used to merge all touching or overlapping rectangles into single polygons (this is why preserving the gaps between intersections is important).

The new polygon feature preserved the merge, truncate, and zone-of-influence logic from the tabular processing, but reduced the number of features/records to one per intersection. At this point, the new intersection features lacked attributes except for geometry and object ID (OID), which ArcMap adds to feature classes automatically. To give each intersection a unique identification, the OID was copied into a new field called ‘IntsctID’ (OID itself is not a stable identifier because it changes when files are modified).

Creating the polygons resulted in several cases where intersection segments were merged inappropriately – mainly at overpasses. However, this set was small enough that errantly merged polygons were readily located (using the select-by-location function to find intersection polygons that overlapped another feature class, ‘PONTUS,’ which inventories Kentucky’s bridges and overpasses). The polygons were manually edited to correct these mismatches. Several additional intersection polygons that represented pseudo-intersections were later deleted. For this and other reasons, attribute data from point and line feature classes were needed to populate the new intersections attribute table. This was accomplished using a process called stamping. To do this, a one-to-many spatial join was executed with the intersection polygons to the points or line segments with the desired data. This stamped each feature in the target with the intersection ID of the polygon it fell within. Desired attributes in the stamped features were then summarized (or dissolved with a ‘Statistics Field’ set to sum by intersection ID). The resulting table was then joined back to intersection polygons with the newly derived data. To maintain clarity, two copies of the intersection polygon feature were created. One called ‘Stamps’ was left as is. The other, called ‘Intersections,’ was augmented with attribute fields from later processing.

2.3.2 Derived Fields and Pseudo-Intersections
The dissolve tool with statistical fields was used extensively on the node features to aggregate data. First, the ‘IntsctNodes’ feature was stamped with the intersection IDs so that each node falling inside one of the intersection polygons adopted the latter’s unique identifier (the ‘IntsctID’ field). ‘IntsctNodes’ was then dissolved by various fields or combinations of fields to count or otherwise summarize other attributes. For example, the nodes were counted in each stack by dissolving for Node ID with a statistics field that counted object IDs (or any other unique identifier). A secondary dissolve was carried out on the derived stack features by intersection ID to count the node stacks that fell within an intersection (in most cases there was only one stack per intersection, but more complicated intersections had several). The number of unique route IDs in one node stack was counted to identify loopbacks – points at which a route curves around to intersect itself. A loopback occurs if the same route ID appears on 3–4 nodes in the same stack.
Counting the number of nodes with the same route ID within an intersection was crucial for calculating the number of approaches per intersection (see below). Any of these counts could then be copied into ‘Intersections’ (the polygon feature) using a one-to-one table join.

For some of the desired statistical characteristics, additional fields were added to the feature class prior to the dissolve. Parts of the 17-character Route ID were split off to create fields for Section ID (the last 3 characters) and LRS-ID (the first 13 characters). Section IDs are three-digit codes that define routes based on their type. For instance, the Y-leg of a Y-intersection is defined by the range of section numbers, 020–029, and a crossover by 030–069. A section number of 000 indicated a standard route. A number of additional ‘count’ fields were added to identify nodes by route types. These were then populated with ones or zeroes depending on whether the section ID fell within the range defined by that category. By definition, each node can only belong to one type. There cannot be more than one of these count fields populated with a one. During the subsequent dissolves, a statistics field was added for each count. This field tabulated the number of nodes that belonged to each section ID category within the dissolve grouping.

Count fields helped identify pseudo-intersections for deletion. Pseudo-intersections are divergence/convergence points on divided highways, isolated crossovers, the divergence point of non-merging Y-intersections, and route-congruent ramp intersections (i.e., intersection of ramps and other routes of the same LRS-ID). Node stacks with three nodes and a ratio of 2:1 between standard route (‘000’) and non-cardinal divided highway route (‘010’) nodes is a divided highway pseudo-intersection. Any three-node stack that contains a crossover has the potential to be part of an isolated crossover. These node stacks were not deleted. They were used to define secondary binary count fields, this time with ones and zeroes in fields to signify different types of pseudo-intersections. Another dissolve of the node stacks by ‘IntsctID’ grouped the node stacks in an intersection with counts of the number of node stacks that were pseudo-intersections. A new calculation field was then added to this table, which subtracted the number of pseudo-intersections from the count of total node stacks. If this field equaled zero (i.e., all of the node stacks in the intersection were pseudo-intersections), it flagged the entire intersection for deletion. An example of a divided highway identified is shown in Figure 1 below.
2.3.3 Route Ranking System and Master Nodes

By definition, intersections include multiple node-points and (except loopbacks) multiple routes. To anchor the intersection to one point, one of the participating routes was chosen as the ‘main route’ of the intersection and one of its associated node points was selected as the index or ‘master node.’ The main route was also used to define certain features for intersection classification (specifically divided versus undivided). To determine which route to use, a ranking system was developed translating the 17 character unique route ID into a 16-digit numerical value. The lower the number, the higher the route rank. In this hierarchy, priority was given to standard routes (with -000 section IDs, as opposed to ramps, non-cardinal divided routes, crossovers, etc) with higher levels of government jurisdiction (state before local, for example). Routes with the same rank according to this criteria were then arbitrarily numbered according route numbers and county numbers, such that no two routes had the same number. Among the routes participating in the intersection, the one with the lowest rank number was defined as the main route. The master node was, in turn, defined as the point record in the intersection associated with the main route which (if the main route passed through more than one node in the same intersection) had the lowest milepoint.

2.3.4 Additional Data

Data for approach counts, main routes, and master nodes were all derived from information already contained in the node table. But additional data were required to classify intersections and to develop the safety performance functions (see below). These included data on traffic controls, traffic volume, directionality (one-way or two-way), crash statistics, jurisdictional level (state or local), and functional classes (including whether the route was designated urban or rural). Various methods were used to aggregate and integrate this data, but the most efficient method typically involved a variation of the stamping method described previously, using either ‘IntsctNodes’ or ‘IntsctSeg’ as intermediaries for the transfer.
The statistics for crashes for 2009–2014 (total crashes and crashes broken down by severity type) were based on a table that summarized the intersections in each segment according to route ID as well as beginning and ending milepoints. Each crash was assigned a date, route, and milepoint. If a crash milepoint fell between the endpoints of a segment it was assigned to that segment, and count fields were generated for the total number of crashes as well as the number of crashes in each severity type (the KABC0 scale was used). The resulting table was linear referenced, stamped with intersection IDs, aggregated by intersection ID with sum of total and subcategory crashes included, and joined with ‘Intersections.’ This and any of the segment-based data (see below) could also be spatially joined to an already-stamped ‘IntsctSeg’ feature class, and summarized by intersection later as one large batch.

AADT and other classificatory data were downloaded from KYTC’s website, drawn either from the fields of the ‘AllRds’ shapefile or from more specialized datasets listed under Highway Information System (HIS). Data were acquired as shapefiles, however, they were not used directly. Instead, DBF tables were extracted and their data re-plotted using the standard version of ‘AllRds_M.’ This step eliminated minor geometric differences found among some routes in the different datasets. These differences result from the Cabinet frequently updating its data. Before the data was re-plotted, some tables were superimposed using the Route Overlay tool based on route and start and end milepoints. The Route Overlay tool superimposed segments defined by each milepoint pair and split any that overlapped into smaller segments. This preserved the attributes of all parent segments.

Re-plotted segments were clipped to intersections and stamped with intersection IDs. To determine if multiple new data segments overlapped one approach segment, a select-for-location was run for segments that touched point features in ‘IntsctNodes.’ If any were found that did not touch an intersection node, they would have been deleted based on the assumption that the part of a route segment closest to the intersection took priority.

Where categories were very simple, selecting-by-location and the field calculator were used rather than directly joining features or attribute tables. For example, the ‘Jurisdiction’ field in ‘Intersections’ was populated by selecting state routes on a route plot, selecting-by-location intersections that touched the selected routes, and entering “state-state” in the field calculator for the selected features. A new selection for local roads was then created in the route feature class and another select-by-location done for ‘Intersections’, but this time with ‘select from current selection’ checked. The result was a smaller selection all mislabeled “state-state.” Recalculating the selected features as “state-local” overwrote this smaller set while leaving the correct “state-state” category untouched. By default, all of the remaining unfilled records in jurisdiction were “local-local.”

The data for intersection controls was derived from a separate KYTC file called route logs. This shared a similar architecture as the node usage file, with points stacked at intersections. However, intersections were allocated one point per route, rather than one per approach. It also encompassed a much smaller subset of intersections than node usage. Like the node usage table, when plotted in ArcMap, points were situated on the wrong side of distant breaks. These were referred to as ‘junction-disjunctions’ in the route log table. A procedure similar to the one described in the section 2.1.2 was used to fix this.

Not every intersection control point lined up with the exact point of an intersection in ‘AllRds_M,’ although most did. Non-aligning points were discarded, but most of these could be salvaged in future iterations of this database by using one or more of the proximity tools in ArcGIS to assign them to the closest intersections. The points that lined up with intersections were integrated into the dataset using table and spatial joins.

The primary information used from route logs came from the field ‘CntrType,’ which classifies traffic controls into a small number of categories including stop signs, traffic lights (signals), and no control. Since route log point features are specific to routes and not whole approaches, it was assumed that where a route constituted two legs the traffic control type applied to both. Route logs were only available for a limited number of intersections, but often not for all of the approaches. This number was stretched by making a couple of assumptions. For signalized intersections, it was assumed that control applied to all legs entering the intersection (whether or not the intersecting route was covered by route
logs or not). Where the route log listed ‘no control’ for a state route intersecting a local route and had no data for the local route, it was assumed that the local route must have a stop control.

2.3.5 Approach Counts

Approach refers to a route through which traffic enters or leaves an intersection. For simple intersections with one node stack, the number of approaches equals the number of records (omitting distance breaks) with the same node ID. If intersections have multiple node stacks, counting approaches is more complicated. For example, a Y-intersection with three node stacks has nine individual nodes, but only three approaches. The segments between the node stacks that make up the intersection are internal to the intersection. As such, they do not constitute approaches. There are far more complicated combinations of node stacks that occur in some intersections, often combining multiple Y-intersections, divided routes, crossovers, and other types of converging routes.

To count the approaches in an intersection, the number of approaches per route was calculated within an intersection. Certain types of routes were considered redundant; that is, they were not separate approaches but merely a sub-component of one that had already accounted for. For example, the cardinal and non-cardinal sides of a divided highway were classified as one approach. This was accounted for by labelling NC routes as having zero approaches. The same logic for Y-intersections and crossover routes was used since they are internal to the intersection and do not generate traffic that enters or leaves that had not been accounted for previously. For all other route types, a route was classified as having one approach if it terminated at the intersection, two approaches if it passed through the intersection, and three if it formed a loopback. At a one-stack intersection, this equaled the number of nodes per route, but at complicated intersections a route might pass though several node stacks, generating two extra nodes for each one it passes through. A route, however, generates only one node on a route it terminates at (i.e., its starting or ending point). Therefore, a route with an odd number of nodes terminates at the intersection, whereas one with an even number of nodes passes through it (correcting for loopbacks).

To count the number of approaches per route, the new fields and dissolve-count statistics described in section 2.3.2 were used. These calculations required us to perform a series of joins and dissolves, which produced a table that counted the number of nodes per route per intersection, with additional fields indicating the presence of loopbacks and identifying redundant routes (any within the range 010–069). A field called ‘ApprRt’ was added to this table indicate the number of approaches for each route. For loopbacks, this equaled ‘3’, and for redundant route types, ‘0’. For all other cases, routes were assigned a ‘1’ if it had an odd number of nodes and a ‘2’ for an even number. A final summary table was then generated for the ‘IntsectID’ field with a field that summed all ‘ApprRt’ numbers for that intersection. This table was joined with the Intersections feature class; the sum of ‘ApprRts’ was copied into a field labeled ‘ApprCnt.’

2.3.6 Calculating Entering Traffic

The volume of traffic entering intersections was calculated using the value of Average Annual Daily Traffic (AADT) for each leg, basing the estimate on whether it was a one- or two-direction leg. The AADT value was obtained from a downloadable HIS shapefile called ‘TF’ (i.e., traffic flow); specifically, this information was contained in a field called ‘Last Count.’ This was available for a limited number of road segments. It contained estimates of road traffic passing through the segment (both ways, except in the case of one-way routes). The Cabinet’s ‘AllRds’ shapefile contains the field ‘TYPE_OP,’ which defines whether a route was one-way, two-way, or part of a divided route. To correct for minor differences between ‘TF’ and ‘AllRds’ and the version of ‘AllRds_M’ the DBF tables extracted from the ‘TF’ file were worked with, combined them using the route overlay tool, and then re-plotted them with our ‘AllRds_M.’

A spatial join assigned data from the new, combined event layer to the segments of the Approaches feature class wherever they overlapped. The ‘TF’ data did not cover all of the intersection segments. For intersections with missing data, an arbitrary proxy value of 300 was assigned. In future
iterations of this database, the hope is to improve estimates of the proxy values, possibly based on the AADT of nearby routes or with county-specific estimates.

Our goal was to estimate the total amount of traffic entering intersections, while the AADT (‘Last_Count’) values from ‘TF’ included traffic in both directions (with the exception of one-ways). Therefore, AADT for two-way segments (or divided segments) were halved, while those for one-ways were either taken in their entirety or reduced to zero. This was contingent on what direction the traffic moved relative to the intersection. The following formulas were used for this calculation:

\[
\begin{align*}
\text{If } \text{TYPE\_OP} = 1 \text{ and } \text{Knu\_S\_E} = 'S', \text{ entering traffic} &= \text{LAST\_COUNT} \times 1 \\
\text{If } \text{TYPE\_OP} = 1 \text{ and } \text{Knu\_S\_E} = 'E', \text{ entering traffic} &= \text{LAST\_COUNT} \times 0 \\
\text{If } \text{TYPE\_OP} = 2 \text{ or } \text{TYPE\_OP} = D, \text{ entering traffic} &= \text{LAST\_COUNT} \times 0.5
\end{align*}
\]

This adjusted AADT figures for each route so they could then be summed together by intersection to gauge how much traffic entered intersections. For the purposes of the Safety Performance Function (SPF), however, this figure was split into two numbers – one for traffic entering via the intersection’s main route (called ‘Major AADT’ in the SPF) and one for traffic entering the intersection from all other routes at the intersection (called ‘Minor AADT’). In the rare event that either of these equaled zero (which could happen if the ‘Last_Count’ field in the original traffic function table was zero, or if the only major or minor route was one-leg one-way leaving the intersection), the number was changed to ‘1’ because the SPF development cannot be include zeroes in the major or minor AADT fields.

3. INTERSECTION DATABASE AND SAFETY PERFORMANCE FUNCTIONS

3.1 Intersection Classification

To develop Safety Performance Functions (SPF), the intersection dataset was divided into 20 categories. Approximately one-third of all intersections had sufficient attribute data to be classified using this scheme. Most of the excluded intersections were local routes that intersected other local routes. Typically, these lacked route-log information or AADT values. Table 1 lists the 20 categories, briefly describes them, and provides a count of how many intersections fell into each category. The category with the most intersections was ‘U3rP,’ which is undivided 3-legged rural intersections with a stop control on at least one leg. Other intersection types, such as ‘D3rS’ of are comparatively rare, and there were only a handful of intersections that met this description in the entire state. There were a total of 182,384 intersections when considering all categories, even those not listed below.
Numerous features were examined before settling on these categories. The classifications were based on three considerations: whether the intersection was divided or undivided; the number of approaches; and the type of traffic control (e.g., full stop, at least partial stop, or signal). These variables existed in 24 combinations; four combinations (all of which were full-stop intersections on divided routes) did not occur in the dataset. This left the 20 remaining classes listed in Table 1.

### 3.2 Safety Performance Functions

SPFs were developed for each of the 20 intersection categories. For a given category, each intersection, its five-year crash history, and the AADT of each approach were compiled in a table. Using the R statistical software program, negative binomial regression analysis was performed twice for each category. The first analysis examined all crashes during the five-year period, while the second looked at only fatal and serious injury crashes (KAB). Regression analysis was performed on the data using the following functional form:

\[
y = e^{a AADT_{maj}^{b1} AADT_{min}^{b2}}
\]

(Equation 1)

Where: \(y\) = predicted number of crashes
\(AADT_{maj}\) = Major approach entering average annual daily traffic
\(AADT_{min}\) = Minor approach entering average annual daily traffic
\(a, b1, b2\) are parameters from the negative binomial regression

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U3rF</td>
<td>undivided 3-leg rural full stop</td>
<td>78</td>
</tr>
<tr>
<td>2</td>
<td>U3rP</td>
<td>undivided 3-leg rural (at least) partial stop</td>
<td>37256</td>
</tr>
<tr>
<td>3</td>
<td>U3rS</td>
<td>undivided 3-leg rural signal</td>
<td>96</td>
</tr>
<tr>
<td>4</td>
<td>U3uF</td>
<td>undivided 3-leg urban full stop</td>
<td>68</td>
</tr>
<tr>
<td>5</td>
<td>U3uP</td>
<td>undivided 3-leg urban (at least) partial stop</td>
<td>10252</td>
</tr>
<tr>
<td>6</td>
<td>U3uS</td>
<td>undivided 3-leg urban signal</td>
<td>583</td>
</tr>
<tr>
<td>7</td>
<td>U4rF</td>
<td>undivided 4+ leg rural full stop</td>
<td>77</td>
</tr>
<tr>
<td>8</td>
<td>U4rP</td>
<td>undivided 4+ leg rural (at least) partial stop</td>
<td>4202</td>
</tr>
<tr>
<td>9</td>
<td>U4rS</td>
<td>undivided 4+ leg rural signal</td>
<td>166</td>
</tr>
<tr>
<td>10</td>
<td>U4uF</td>
<td>undivided 4+ leg urban full stop</td>
<td>89</td>
</tr>
<tr>
<td>11</td>
<td>U4uP</td>
<td>undivided 4+ leg urban (at least) partial stop</td>
<td>2484</td>
</tr>
<tr>
<td>12</td>
<td>U4uS</td>
<td>undivided 4+ leg urban signal</td>
<td>1492</td>
</tr>
<tr>
<td>13</td>
<td>D3rP</td>
<td>divided 3-leg rural (at least) partial stop</td>
<td>729</td>
</tr>
<tr>
<td>14</td>
<td>D3rS</td>
<td>divided 3-leg rural signal</td>
<td>26</td>
</tr>
<tr>
<td>15</td>
<td>D3uP</td>
<td>divided 3-leg urban (at least) partial stop</td>
<td>1292</td>
</tr>
<tr>
<td>16</td>
<td>D3uS</td>
<td>divided 3-leg urban signal</td>
<td>335</td>
</tr>
<tr>
<td>17</td>
<td>D4rS</td>
<td>divided 4+ leg rural (at least) partial stop</td>
<td>459</td>
</tr>
<tr>
<td>18</td>
<td>D4rS</td>
<td>divided 4+ leg rural signal</td>
<td>66</td>
</tr>
<tr>
<td>19</td>
<td>D4uP</td>
<td>divided 4+ leg urban (at least) partial stop</td>
<td>560</td>
</tr>
<tr>
<td>20</td>
<td>D4uS</td>
<td>divided 4+ leg urban signal</td>
<td>832</td>
</tr>
</tbody>
</table>
The following code was used to generate the needed parameters:

```r
# To read the data into R
data = read.csv("C:/RevSPF.csv", header=T)
attach(data)

# Use this to specify a subset of data (comment out this and the next line if unneeded)
data <- subset(data, ClassNum==1)
attach(data)

# Define variables (comment and uncomment based on severity)
crash = (All_Crashes)
# crash = (KAB_Crashes)
logMajADT = log(MajAADT)
logMinADT = log(MinAADT)

# To load the MASS library which contains the command glm.nb
library(MASS)

# To generate the negative binomial models
init.theta = 3
model1 = glm.nb(crash ~ logMajADT + logMinADT, init.theta = init.theta)  # Fitting

# To retrieve the summary of the models
summary(model1)
```
Table 2 summarizes the output of the regression equation for each intersection type.

**Table 2 SPF Parameters**

<table>
<thead>
<tr>
<th>Type</th>
<th>Intercept</th>
<th>Log_Maj</th>
<th>Log_min</th>
<th>Theta</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Crashes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-6.8508</td>
<td>0.7453</td>
<td>0.2635</td>
<td>0.86</td>
</tr>
<tr>
<td>2</td>
<td>-10.0896</td>
<td>0.84818</td>
<td>0.58105</td>
<td>0.854</td>
</tr>
<tr>
<td>3</td>
<td>-6.78328</td>
<td>0.89225</td>
<td>0.15249</td>
<td>1.95</td>
</tr>
<tr>
<td>4</td>
<td>-2.40847</td>
<td>0.23563</td>
<td>0.25905</td>
<td>2.254</td>
</tr>
<tr>
<td>5</td>
<td>-6.58266</td>
<td>0.70294</td>
<td>0.29223</td>
<td>0.9444</td>
</tr>
<tr>
<td>6</td>
<td>0.13917</td>
<td>0.23507</td>
<td>0.10347</td>
<td>2.013</td>
</tr>
<tr>
<td>7</td>
<td>-8.17309</td>
<td>1.1404</td>
<td>0.08278</td>
<td>2.74</td>
</tr>
<tr>
<td>8</td>
<td>-9.69662</td>
<td>0.74447</td>
<td>0.72347</td>
<td>1.0202</td>
</tr>
<tr>
<td>9</td>
<td>-1.19747</td>
<td>0.20673</td>
<td>0.25183</td>
<td>1.895</td>
</tr>
<tr>
<td>10</td>
<td>-2.17487</td>
<td>0.19232</td>
<td>0.35288</td>
<td>1.363</td>
</tr>
<tr>
<td>11</td>
<td>-1.72875</td>
<td>0.2612</td>
<td>0.21699</td>
<td>0.9774</td>
</tr>
<tr>
<td>12</td>
<td>1.26001</td>
<td>0.11229</td>
<td>0.14798</td>
<td>1.6518</td>
</tr>
<tr>
<td>13</td>
<td>-4.97683</td>
<td>0.24124</td>
<td>0.59428</td>
<td>0.7245</td>
</tr>
<tr>
<td>14</td>
<td>-2.6982</td>
<td>0.4885</td>
<td>0.1142</td>
<td>1.945</td>
</tr>
<tr>
<td>15</td>
<td>1.61303</td>
<td>-0.66379</td>
<td>0.15371</td>
<td>0.6737</td>
</tr>
<tr>
<td>16</td>
<td>0.5548</td>
<td>0.20884</td>
<td>0.11222</td>
<td>1.948</td>
</tr>
<tr>
<td>17</td>
<td>-6.38656</td>
<td>0.25745</td>
<td>0.79386</td>
<td>0.891</td>
</tr>
<tr>
<td>18</td>
<td>-1.21364</td>
<td>0.18135</td>
<td>0.32262</td>
<td>3.57</td>
</tr>
<tr>
<td>19</td>
<td>-1.90987</td>
<td>0.26754</td>
<td>0.2528</td>
<td>1.0849</td>
</tr>
<tr>
<td>20</td>
<td>1.16855</td>
<td>0.15029</td>
<td>0.15579</td>
<td>1.8189</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Intercept</th>
<th>Log_Maj</th>
<th>Log_min</th>
<th>Theta</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAB Crashes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-7.52232</td>
<td>0.74916</td>
<td>0.07044</td>
<td>0.225</td>
</tr>
<tr>
<td>2</td>
<td>-11.8225</td>
<td>0.77761</td>
<td>0.63695</td>
<td>0.6706</td>
</tr>
<tr>
<td>3</td>
<td>-8.29651</td>
<td>0.92564</td>
<td>-0.06765</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>-5.8866</td>
<td>0.1858</td>
<td>0.3356</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>-10.5728</td>
<td>0.8795</td>
<td>0.28738</td>
<td>0.9064</td>
</tr>
<tr>
<td>6</td>
<td>-3.22177</td>
<td>0.29948</td>
<td>0.11612</td>
<td>1.466</td>
</tr>
<tr>
<td>7</td>
<td>-14.6596</td>
<td>1.75784</td>
<td>-0.04074</td>
<td>0.51</td>
</tr>
<tr>
<td>8</td>
<td>-12.541</td>
<td>0.68717</td>
<td>0.93473</td>
<td>0.755</td>
</tr>
<tr>
<td>9</td>
<td>-9.8758</td>
<td>0.6791</td>
<td>0.5097</td>
<td>1.834</td>
</tr>
<tr>
<td>10</td>
<td>-2.24074</td>
<td>-0.06297</td>
<td>0.24041</td>
<td>0.676</td>
</tr>
<tr>
<td>11</td>
<td>-6.95671</td>
<td>0.51766</td>
<td>0.30547</td>
<td>0.7413</td>
</tr>
<tr>
<td>12</td>
<td>-1.6062</td>
<td>0.14811</td>
<td>0.14768</td>
<td>1.234</td>
</tr>
<tr>
<td>13</td>
<td>-8.8937</td>
<td>0.4526</td>
<td>0.6505</td>
<td>0.498</td>
</tr>
<tr>
<td>14</td>
<td>-4.6441</td>
<td>0.4695</td>
<td>0.1449</td>
<td>1.43</td>
</tr>
<tr>
<td>15</td>
<td>-3.51331</td>
<td>0.19135</td>
<td>0.20701</td>
<td>0.5931</td>
</tr>
<tr>
<td>16</td>
<td>-2.54404</td>
<td>0.26389</td>
<td>0.12154</td>
<td>2.538</td>
</tr>
<tr>
<td>17</td>
<td>-11.1975</td>
<td>0.5557</td>
<td>0.8629</td>
<td>0.577</td>
</tr>
<tr>
<td>18</td>
<td>-2.52751</td>
<td>0.15942</td>
<td>0.22736</td>
<td>1.872</td>
</tr>
<tr>
<td>19</td>
<td>-7.80787</td>
<td>0.57463</td>
<td>0.36916</td>
<td>1.348</td>
</tr>
<tr>
<td>20</td>
<td>-0.97881</td>
<td>0.12115</td>
<td>0.13871</td>
<td>1.794</td>
</tr>
</tbody>
</table>
4. CONCLUSIONS

This study produced a comprehensive, fully updateable intersection database. Adopting the node usage table as database's foundation ensured that the location information in the intersection database is able to reflect any updates in roadway geometry that may cause adjustments in downstream milepoints. The database will be revised regularly to reflect any edited, removed, or added nodes in the node usage table. Each intersection ID was indexed to one or more node IDs. Comparing the current node usage table to previous versions will let users identify any changes, flag them, and apply the appropriate correction to the associated intersection.

Using the SPFs developed as part of this study will provide a more data-driven approach to decision making related to intersection safety than the previous critical rate analysis. It is anticipated that accounting for the effects of regression-to-the-mean and ensuring that low-volume intersections are not ranked artificially high will assist KYTC in deciding where to prioritize the installation of intersection safety countermeasures.

There are many ways in which the processes described in this report could be refined and expanded in future versions of the database. A major limitation with the current version is the absence of AADT data and traffic control information for many (in fact, most) of the intersections in the larger database. It has been suggested that existing county-specific data for Vehicle Miles Travelled (VTM) could serve as a proxy to estimate AADT when exact measures are unavailable. Future iterations of the process will likely integrate interchanges as an analytical category. Preliminary work also suggests that the skew angles between intersection approaches could be meaningfully integrated into the intersection classifications. Furthermore, it is anticipated that once the database and associated SPF results are made available to expert users, additional improvements will be suggested. As such, a quality control protocol will be implemented to ensure that errors in the database are addressed each year.
REFERENCES


