CALIBRATING DYNAMIC CAR-FOLLOWING MODEL USING VEHICLE TRAJECTORY DATA: A DYNAMIC TIME WARPING APPROACH

Jeffrey Taylor
Department of Civil and Environmental Engineering
University of Utah, Salt Lake City, UT 84112-0561
Email: jeff.d.taylor@utah.edu

Xuesong Zhou*, Ph.D.
Department of Civil and Environmental Engineering
University of Utah, Salt Lake City, UT 84112-0561
110 Central Campus Dr., MCE 2035
Salt Lake City, Utah, 84112-0561
Phone: (801) 585-6590
Email: zhou@eng.utah.edu

Nagui M. Rouphail, Ph.D.
Director, Institute for Transportation Research and Education (ITRE)
Professor of Civil Engineering
North Carolina State University
Centennial Campus, Box 8601
Raleigh, NC 27695-8601
Tel.: (919) 515-1154
Email: rouphail@eos.ncsu.edu

*Corresponding Author

Submitted August 1, 2011
Revised November 14, 2011

5500 words + 7 figures + 1 table = 7500 words

Submitted for presentation at the 91th Annual Meeting of the Transportation Research Board, January 2012, Washington, D.C.

ABSTRACT

After first extending Newell's car-following model to incorporate time-dependent parameters, this paper describes the Dynamic Time Warping (DTW) algorithm and its application for calibrating this microscopic simulation model by synthesizing driver trajectory data. Using the unique capabilities of the DTW algorithm, this paper briefly attempts to examine driver heterogeneity in car-following behavior, as well as the driver's heterogeneous situation-dependent behavior within a trip, based on the calibrated time-varying reaction times and critical jam spacing. The standard DTW algorithm is enhanced to address a number of estimation challenges in this specific application, and a numerical experiment is presented with vehicle trajectory data extracted from the Next Generation Simulation (NGSIM) project for demonstration purposes. The DTW algorithm is shown to be a reasonable method for processing large vehicle trajectory datasets, but requires significant data reduction to produce reasonable results when working with high resolution vehicle trajectory data. Additionally, singularities present an interesting match solution set to potentially help identify changing driver behavior; however, they must be avoided to reduce analysis complexity.

Keywords: Dynamic Time Warping, Car-following model, Driver behavior heterogeneity, Vehicle trajectory data

1. INTRODUCTION

It has been evident over the past decade that continuing advances in traffic simulation systems intrinsically depend on advances in driving behavior modeling. Traffic incidents, driver inattention, and recurring bottlenecks are widely recognized as the most significant factors affecting congestion (in conjunction with bad weather, work zones and poor signal timing). Thus, for most planning and operational applications, transportation analysts must strive to achieve realistic representations of driving behaviors during unusual and complex traffic situations.

Individual driver characteristics, such as reaction time and spacing in jam conditions, are expected to be altered during complex driving conditions, and thus are expected to contribute toward the occurrence of stop-and-go traffic patterns which eventually translate into changes in travel time reliability, roadway capacity, and vehicle emissions. This paper aims to address the following theoretically challenging and practically important question: How can we enhance existing traffic simulation models and the corresponding calibration algorithms so that they can more accurately and sensitively capture the impacts of time-dependent and situation-dependent driver behavior?

To examine such issues, this paper attempts to extend Newell's classical simplified linear car-following model (*1*) to incorporate time-dependent reaction time and spacing parameters. This enhanced microscopic simulation model will be calibrated through a novel use of the Dynamic Time Warping (DTW) algorithm using recently available continuous or semi-continuous vehicle location, speed and acceleration data across different time periods. Finally, a numerical experiment is presented to explore the capabilities of the DTW algorithm for estimating time-dependent driver behavior under different conditions.

Background

Driving Behavior and Microscopic Simulation Models

Driving behavior has been one of the most difficult human decision-making processes to model. A wide range of car-following models, psycho-physical models and multi-phase traffic flow theories have been proposed in an attempt to capture the driving behavior at a microscopic level. In many existing simulations of driving models, the behavior of a driver is mainly determined by the relative headway, gap, speed and acceleration of the lead and surrounding vehicles. Although a number of traffic simulation models have considered multiple driver classes to accurately describe heterogeneous perception and preferences, most widely used models still assume the same behavioral characteristics under both congested and uncongested driving situations.

Several recent studies have examined driver behavior heterogeneity and its impact on microscopic simulation models. Ossen and Hoogendoorn (*2*) used high resolution trajectory data from a helicopter to find optimal sensitivity and reaction time parameters for individual drivers and for multiple car-following models. A later study by the same authors showed that driver heterogeneity could not be explained only with model parameters, but must also include model specifications (*3*). Most recently, Ossen and Hoogendoorn (*4*) studied vehicle trajectories for passenger vehicles and trucks in different leader-follower scenarios. Their results showed large

variations between how passenger car drivers react to different stimuli and which stimuli influence their behavior, while truck drivers maintained more consistent speeds over time. They also showed that driver behavior can change depending on the leader's vehicle type. Kim and Mahmassani (*5*) calibrated multiple car-following models using NGSIM trajectory data to examine the effects of considering correlation between model parameters. Their results indicate a statistically-significant difference between correlated and uncorrelated parameter sets for the models tested, but the effects varied widely between those models. Laval and Leclercq (*6*) presented a theory for modeling aggressive and timid driver behavior which describes traffic oscillations and their transformation into stop-and-go waves. They specifically identified traffic oscillations as a consequence of drivers' heterogeneous reactions to deceleration waves, but aggressive and timid driving behavior alone could not produce the observed traffic oscillations.

Newell's Car-Following Model

Newell proposed a simplified linear car-following model (LCF) (*1*,*7*) which considers the following two driving modes: (i) Under uncongested conditions, vehicles are driving at free-flow speed, and (ii) Under congested conditions, a following vehicle changes speeds to maintain a minimum jam spacing and a reaction time lag with respect to the leading vehicle's trajectory. Brockfeld et al. (*8*) calibrated and validated a number of well-known car following models, and Newell's simplified LCF model showed reasonable performance with limited calibration efforts.

2. MODEL FORMULATION

Beginning with Newell's Car-Following Model (*1*) under congested conditions, the position of the following vehicle is determined by a space and time offset (d_n and τ_n, respectively) from the lead vehicle, as shown in Eq. 1. Based on a triangular flow-density diagram in FIGURE 1, the LCF can be shown its equivalence with a simplified kinematic wave model: the critical jam spacing is the inverse of jam density. $d_n = 1/k_{jam}$, and the reaction time τ_n is a function of the backward wave speed w; $\tau_n = d_n/w$.

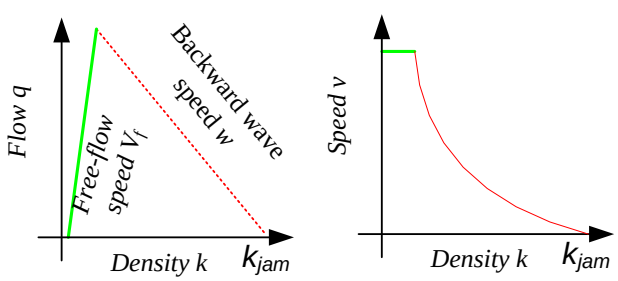


FIGURE 1 Triangular fundamental diagram: a typical value of w is 19 kph and K_{jam} is 112 vehicles/km/lane, which leads to d_n = 9 meters, τ_n = 1.7 seconds

A graphical representation of the model is shown in FIGURE 2 below. This model assumes that τ_n and d_n vary between drivers, but the parameters are constant over time.

$$x_n(t + \tau_n) = x_{n-1}(t) - d_n \quad (1)$$

Ahn, Cassidy, & Laval (*9*) verified Newell's CF model, confirming that τ_n and d_n are different for each driver, but did not explicitly verify that each driver's τ_n and d_n are maintained over their entire trip. As discussed before, we begin our investigation with the hypothesis that τ_n and d_n vary with time to account for situation-dependent car-following parameters, resulting in the formulation shown in Eq. 2 below.

$$x_n(t + \tau_{n,t}) = x_{n-1}(t) - d_{n,t} \tag{2}$$

Newell described his model using the piecewise linear approximation for vehicle trajectories, as shown in FIGURE 2 below. If the velocity of the (n-1)th car and the nth car are the same, the piecewise approximation is reasonable, but this may not be the case where the drivers' reactions to the change in speed are not homogeneous. Our approach does uses a piecewise linear approximation for part of the numerical experiment in Section 4 to match inflection points. However, alternative approaches may be used to investigate the more complicated behavior that takes place during these behavioral responses to stimuli from the leader. To examine the capabilities of the DTW algorithm, we also attempt the numerical experiment without piecewise linear approximation.

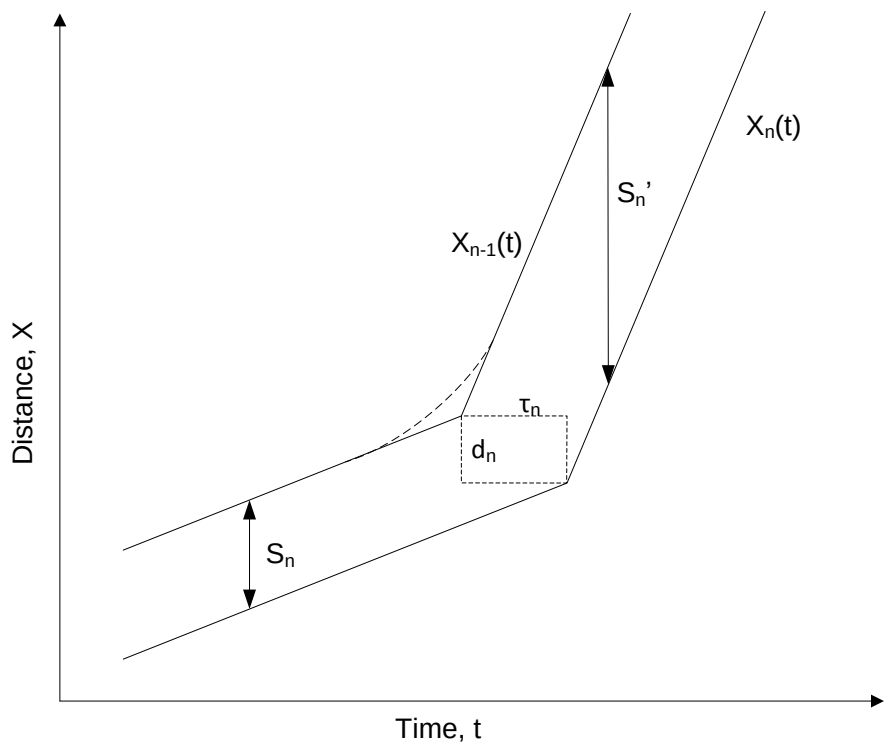


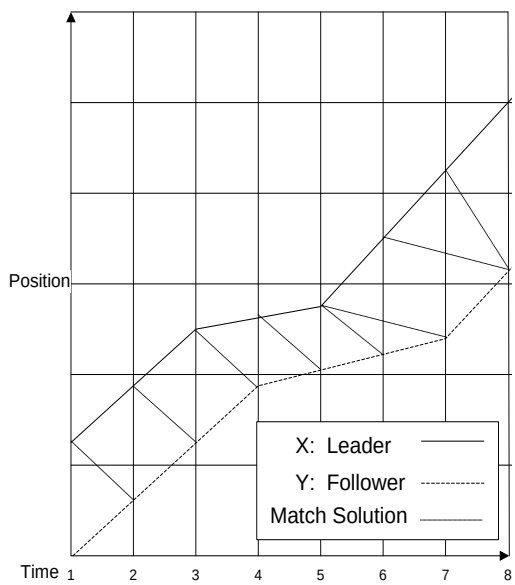
FIGURE 2 Newell's Car-Following Model graphical representation with piecewise linear approximation, replicated from Newell (*1*)

3. DYNAMIC TIME WARPING (DTW) ALGORITHM & ILLUSTRATIVE EXAMPLE

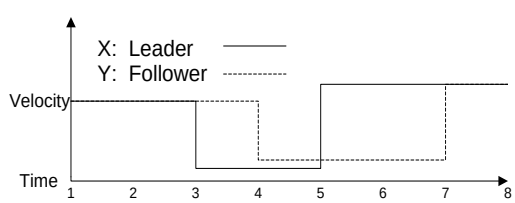
In order to calibrate this extended version of Newell's Car-Following Model, we apply the DTW algorithm to high-resolution continuous or semi-continuous vehicle trajectory data (after data

reduction) in order to identify the space-time points corresponding to the follower's response to the leader's stimulus.

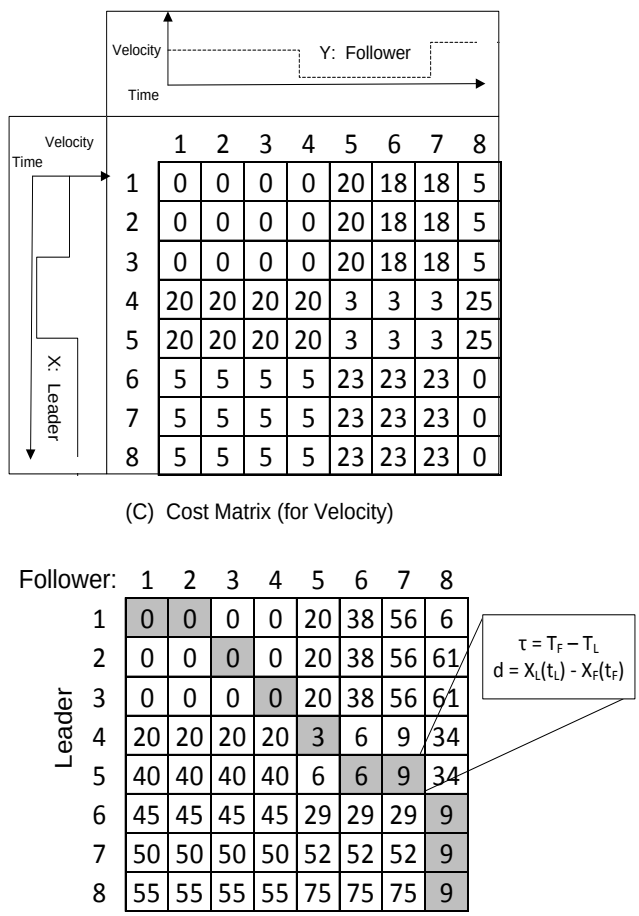
Dynamic Time Warping is an algorithmic approach used to find the optimal alignment between two time-dependent data series. In its basic form, the DTW algorithm first assesses the cost for aligning each data point in one series to all other points in the second series, creating a cost matrix. It then begins at the first data series pair in the cost matrix, calculating the cumulative least cost for continuously moving from the first pair to the last pair in the matrix, creating a cumulative cost matrix. Lastly, the algorithm finds the alignment with the least cumulative cost, called the warp path, using a shortest path algorithm which starts at the last pair and works back to the first pair in the cumulative cost matrix. The cost is a quantitative measure of the similarity or difference between two points (distance is commonly used), but it is also a flexible term which may be modified to suit the application. The algorithm is described in greater detail below as an illustrative step-by-step example to guide the reader. The summary provided here is made in reference to Pavel (*10*) and Keogh et al. (*11*), which may provide further explanations of the DTW algorithm for the reader.



(A) Vehicle Trajectories with DTW Match Solution



(B) Vehicle Velocity Time Series



	1	2	3	4	5	6	7	8
1	0	0	0	0	20	18	18	5
2	0	0	0	0	20	18	18	5
3	0	0	0	0	20	18	18	5
4	20	20	20	20	3	3	3	25
5	20	20	20	20	3	3	3	25
6	5	5	5	5	23	23	23	0
7	5	5	5	5	23	23	23	0
8	5	5	5	5	23	23	23	0

(C) Cost Matrix (for Velocity)

	1	2	3	4	5	6	7	8
1	0	0	0	0	20	38	56	6
2	0	0	0	0	20	38	56	61
3	0	0	0	0	20	38	56	61
4	20	20	20	20	3	6	9	34
5	40	40	40	40	6	6	9	34
6	45	45	45	45	29	29	29	9
7	50	50	50	50	52	52	52	9
8	55	55	55	55	75	75	75	9

(D) Cumulative Cost Matrix with Highlighted Warp Path

FIGURE 3 Illustrative example of DTW algorithm

Input Data

Suppose we begin with two simple vehicle trajectory datasets, as shown in Figure 3(A) below (but without the match solution). This dataset includes a series of position *P*, velocity *V*, and acceleration *A* values corresponding to a series of timestamps *T* for when those data points were recorded. We extract the speed data to produce the time series datasets, X (leader) and Y (follower), with lengths *N* and *M* and indices *i* and *j*, respectively. Velocity data are used in the illustrative example for demonstration purposes, but acceleration data may also be used. These datasets are mathematically described in Eq. 3. The illustrative example uses a relatively simple velocity time series with two speed changes, as shown graphically in Figure 3(B), but the follower velocity is slightly greater than the leader velocity after the first speed change.

$$X = x_1, x_2, \dots, x_N \ \ (Time\ Series\ 1), \quad Y = y_1, y_2, \dots, y_M \ \ (Time\ Series\ 2) \tag{3}$$

Cost Matrix

After extracting the input data for the algorithm, the first step in the algorithm is to develop a cost matrix. The cost of aligning or matching two points is defined by the cost matrix *C*, where the cost for aligning X_i and Y_j is $C(X_i, Y_j)$. The distance between the two points, such as the Euclidean distance, is a commonly chosen measure for the cost matrix. Since the distance is only in one dimension, the Euclidean distance becomes the absolute difference between the two points, as shown in Eq. 4.

$$C(X_i, Y_j) = \sqrt{(X_i - Y_j)^2} = |X_i - Y_j| \tag{4}$$

The cost matrix for the illustrative example is shown in Figure 3(C) with the graphical representation for the leader and follower vehicles presented alongside their respective dimensions. Combining the graphical representations and the cost matrix provides a visual comparison demonstrating how the cost matrix identifies similar points in the two time series datasets. The first plateau ends near $t = 3$ for the leader and $t = 4$ for the follower, which is shown in the cost matrix as zero cost because they are traveling at the same velocity during those time periods. The cost abruptly changes when attempting to compare the second speed plateau for the leader with the first speed plateau for the follower, as shown at (1,4) in the cost matrix.

Cumulative Cost Matrix & Warp Path

The next step is to find the warp path *W*, which is the path between the end points for each dataset with the lowest cumulative cost. We use *k* as the index for the warp path, where *L* is the length of the warp path.

$$W = w_1, w_2, \dots w_L \qquad w_k = (i, j) \tag{5}$$

The warp path must satisfy the following three constraints:

- Boundaries: The start and end points of the datasets must be the start and end points of the warp path. $w_1 = (1, 1) \quad w_L = (N, M)$
- Continuity: The warp path cannot step forward more than one time index in any direction at one time. $w_k = (i, j) \quad w_{k-1} = (i', j') \qquad i - i' \leq 1 \qquad j - j' \leq 1$

- Monotonicity: The warp path must continuously step forward from beginning to end; the algorithm cannot step backward. $w_k = (i,j) \quad w_{k-1} = (i',j') \quad i - i' \geq 0 \quad j - j' \geq 0$

We must use these constraints when creating the cumulative cost matrix D, which describes the least cost required to arrive at any location along any warp path. Starting from the first pair (i = 1, j = 1), the algorithm compares the three possible previous steps in the warp path, looking for the least cost step, and adding to this the cost of the current step. Eq. 6 describes the expression used to construct the cumulative cost matrix.

$$D(i,j) = C(i,j) + \min\big(D(i-1,j-1), D(i-1,j), D(i,j-1)\big) \qquad i \in N,\ j \in M \tag{6}$$

For the illustrative example, the cumulative cost matrix is shown in Figure 3(D). Once the cumulative cost matrix is known, a standard shortest path or dynamic programming algorithm can be adapted used to find the optimal warp path, starting at the end and working backwards to the beginning. The optimal warp path for the illustrative example is highlighted in Figure 3(D), and the matching solution is plotted in Figure 3(A) with the original dataset.

Output Data

The DTW algorithm provides the matching/mapping points between two vehicle trajectories based on the cost of matching those points. These matching points refer directly to the warp path W, which is composed of the matrix locations (i, j) in Eq. 5. The original dataset included datasets for position P, velocity V, and acceleration A for the leader and follower (denoted by L and F subscripts). Each point also has an associated time stamp T, and all datasets are the same length (N or M). The indices i and j from the warp path can then be used to extract the matching time and position data points for the leader and follower from the original datasets. An example of the DTW algorithm's match solution (without data reduction) for two vehicle trajectories is plotted in FIGURE 4 below. Table 1 demonstrates the relationship between the extracted value and the indices stored in the warp path.

TABLE 1 Example of leader's original trajectory dataset and its relation to the index *i* for matching point extraction

	***i* = 1**	***i* = 2**	***i* = 3**	***i* = 4**
Time (sec)	12	12.1	12.2	12.3
Position (ft)	1590	1592.2	1594.4	1496.6
Velocity (mph)	15	15	15	15
Acceleration (ft/s^2)	0	0	0	0

We can further use these extracted points to calculate the τ_n and d_n values for each matching pair k with Eqs. 7 & 8, which can then be used to calculate w_n. The matches in the warp path are made based on the follower's reaction to the leader, so the k index for the warp path becomes the

index for the car following parameters τ_n and d_n as they change over the time duration of the trajectory.

$$\tau_n(k) = T_F(j) - T_L(i) \tag{7}$$

$$d_n(k) = X_L(i) - X_F(j) \tag{8}$$

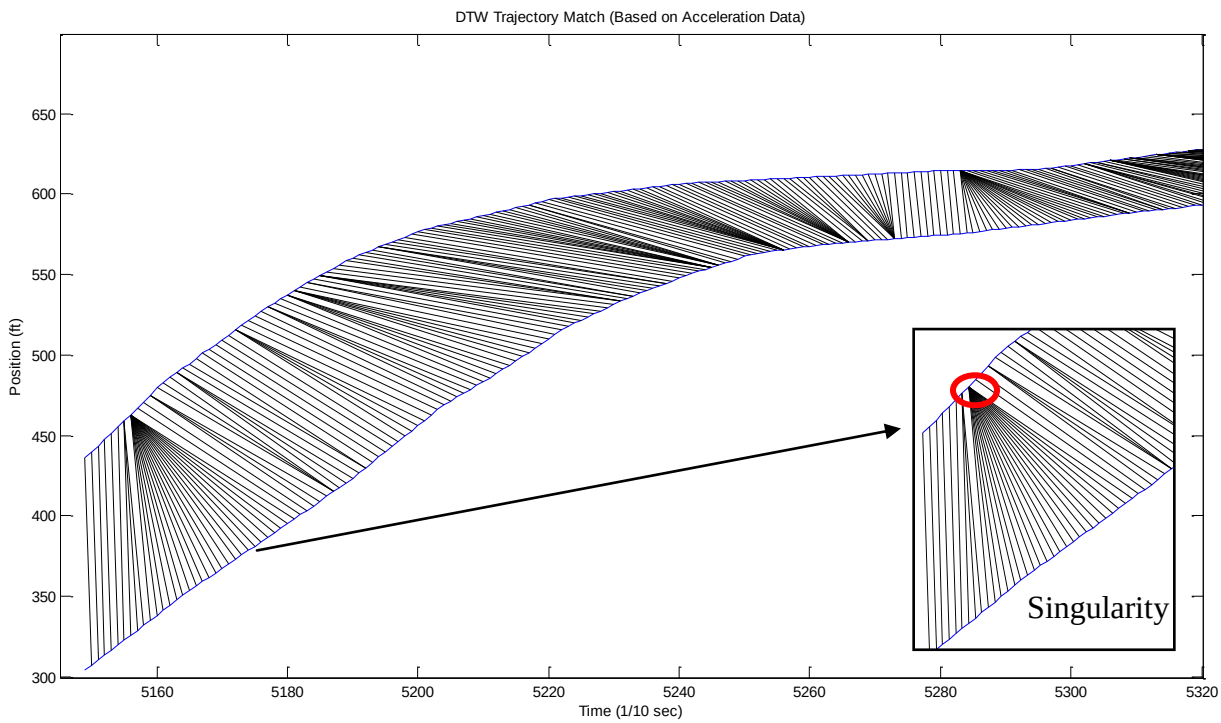


FIGURE 4 Example output for DTW Vehicle Trajectory Match with highlighted singularity

Several potential benefits result from using the DTW algorithm, which are summarized below.

- Time warping produces different reaction times corresponding to different mappings
- Critical spacing is recovered directly from the mappings
- Time warping efficiently processes large amounts of repeated observations of the same drivers over long periods of time at different driving situations

However, DTW also raises several difficulties for estimating time-dependent driving behavior parameters.

- Many possible mappings (solutions) exist: the path-finding algorithm finds the optimal matching result for a given cost matrix that involves measurement errors and model assumption errors; it might not be the most intuitive or likely result. Different solutions can have similar objective function values that are very close to the found "optimal" value

- Measurement errors (locations, speed, acceleration) may influence matching solution quality
- Feature extraction (data reduction) is required to identify inflection points

Algorithm Enhancements for Vehicle Trajectories

While the DTW algorithm was designed to match time series data, this method does require a few modifications to constrain the matching solution within theoretically-acceptable bounds based on our understanding of car-following behavior. First, we can set lower bounds for the car-following parameters τ_n and d_n for acceptable matches ($\tau_n > 0$, $d_n > 0$). If we were to set upper bounds for these parameters, this would be similar to the windowing method described in the following section. However, an upper bound may artificially prevent the algorithm from identifying correct matches, depending on the leader-follower relationship. For example, an abnormally long following distance may exclude several matches for a leader-follower pair. For a conservative approach only implementing the lower bound, the lower bound constraint can be applied when calculating the cost matrix by assigning an artificially-high cost to any matching pairs which violate the constraint.

$$\tau = j - i;\ \ d = x_L(i) - x_F(j); \qquad If\ \tau \leq 0\ OR\ d \leq 0,\ \ c(i,j) = \ 1000;$$

This constraint encourages the algorithm to make theoretically-acceptable matches, but allows unacceptable matches when necessary to guarantee a continuous path for both trajectories.

The shortest path algorithm also requires some minor modifications to account for situations with no obvious best choice for the next step in the warp path. This occurs where the next step cost is the same in more than one available direction. A good example of this situation is a location where an unacceptable match is necessary (where the optimal warping path passes through an unacceptable match point). In these cases, a pre-specified path step direction may be the simplest option (a diagonal step is usually preferred), but may cause issues when the same situation arises consecutively because the shortest path is unknown beyond the current location in the cumulative cost matrix. This issue could be solved by changing the search pattern for the shortest path algorithm to look ahead beyond the next available path step, with a bias toward a diagonal path between the current and future best path points.

4. NUMERICAL EXPERIMENT WITH NGSIM DATA

Starting with data from the NGSIM project for I-80 in California (*15*), we use the DTW algorithm to extract the optimal match points and analyze individual drivers' car-following parameters as they change over time. For this numerical experiment, we've extracted a series of vehicles from Lane 4, including both trucks and passenger cars. Data for these vehicles are available at 0.1 second resolution, and the dataset was trimmed to approximately 60 seconds so that data was available for all vehicles for each time index. The space-time plots for these vehicle trajectories are shown in Figure 5 below. The DTW algorithm was applied using the calculated acceleration to develop the cost matrix, with a lower bound constraint applied when calculating the cost matrix (i.e. artificial cost = 100). The methodology was applied with and without reduced input data. All DTW calculations and visualizations were performed with MATLAB.

Analysis Results With Data Reduction

First, we apply the DTW algorithm to a time series which has undergone data reduction. This was performed manually, where best judgment was used to form a piecewise linear approximation for each vehicle trajectory. The algorithm produced nine match points for the first following vehicle (the truck, highlighted in red in Figure 5), and ten match points for the second following vehicle (the vehicle following the truck). The figure only shows six and seven plotted matches for the first and second following vehicles, respectively, because only acceptable solutions are plotted (i.e. $\tau > 0$, $d > 0$). Complete results for the matches are shown in the time series in Figure 6.

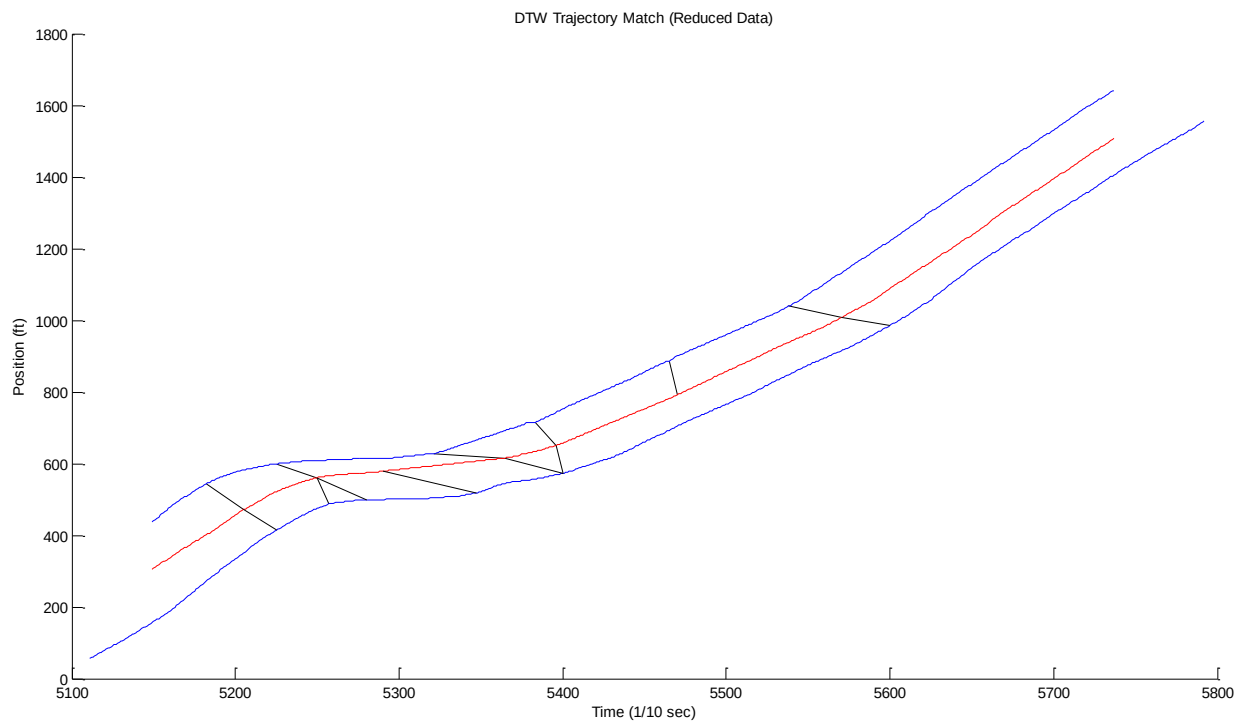


FIGURE 5 DTW trajectory match for reduced data

The matching solution results, especially in the congested region between t = 5150 and t = 5400, appears to show consistent backward wave speeds in multiple locations along the trajectory. Additionally, the wave speed also appears to change in the deceleration and acceleration regions, showing a slight trend toward decreasing before congestion and increasing after congestion. This presents the possibility that situation-dependent car-following parameters may exist, but does not conclusively prove or disprove their existence. Further study on a larger scale is required to investigate these characteristics. Further enhancements to the DTW algorithm which could also improve solution quality (which are not applied here) are discussed in the next section.

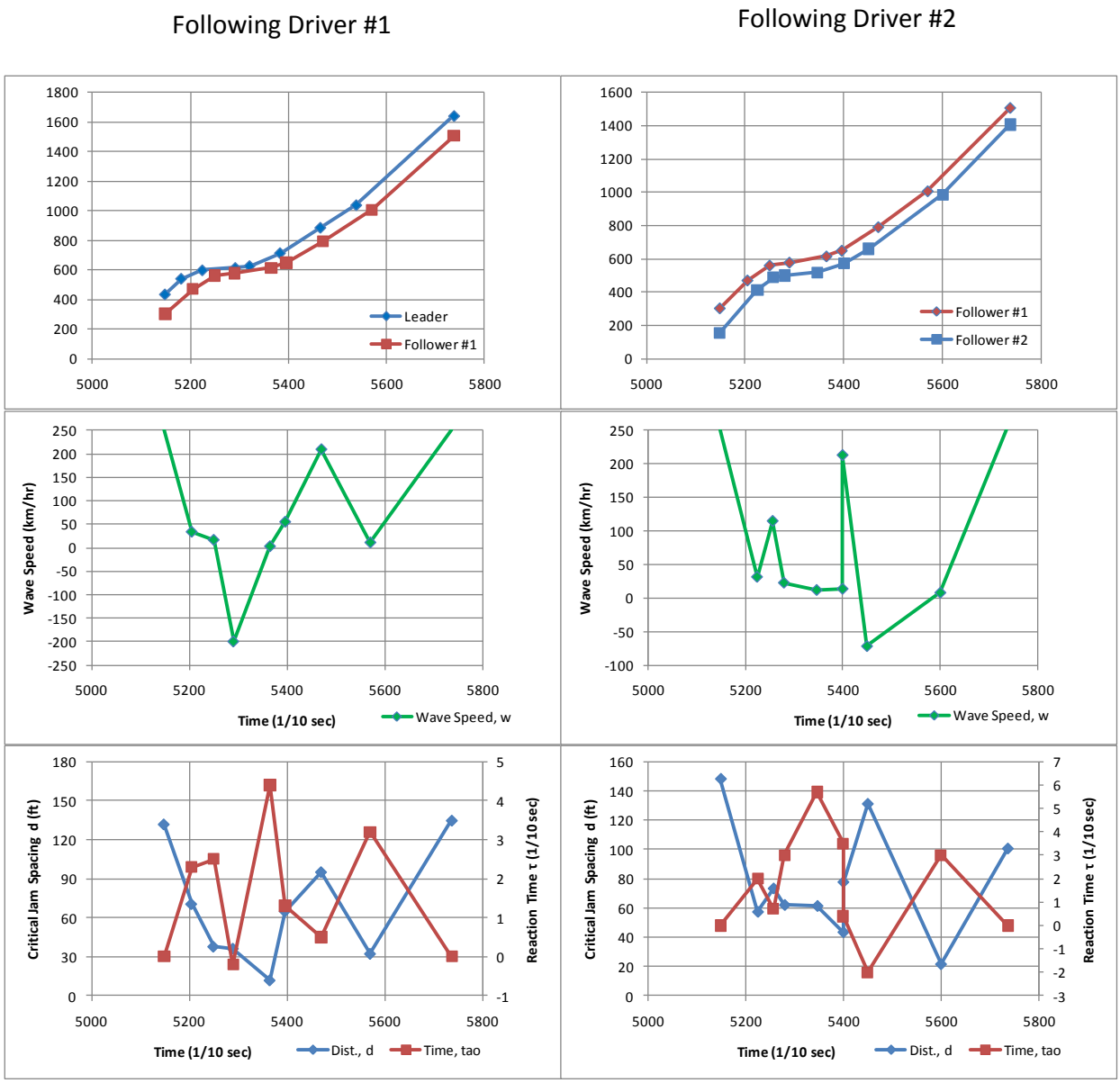


FIGURE 6 Time-series plots for car-following parameters d, τ, and w

Examining the complete solution set in Figure 6, we observe multiple solutions which are not within the boundary constraints ($\tau > 0$, $d > 0$) for the car-following model. The wave speeds at the end points (w = infinity) are ignored because the matching points create vertical lines with $\tau = 0$. We also observe that singularities are located near points of extreme results in the time-series plots, but their influence at these locations is not clear from the simple analysis provided in this numerical experiment.

Without Data Reduction

The datasets are approximately 60 times larger without using the data reduction algorithm (approximately 600 data points for each trajectory). This produces a very different matching

pattern compared to the reduced data matching solution. The match solution in FIGURE 7 is focused upon an area within the plot shown in FIGURE 5 so that the reader may inspect the solution quality. The changing slope is observed again here, indicating a changing wave speed. This plot shows a much stronger case for situation dependent parameters, where the reaction time increases greatly for the second following vehicle after congestion begins and increases after congestion ends. It appears that the matching solutions for both drivers align well with each other in some regions of the plot. However, singularities once again introduce an element of uncertainty in the matching solution. This uncertainty limits our ability to draw conclusions at this state in our research. Plotted results for the time-series car-following parameters are not shown due to space constraints.

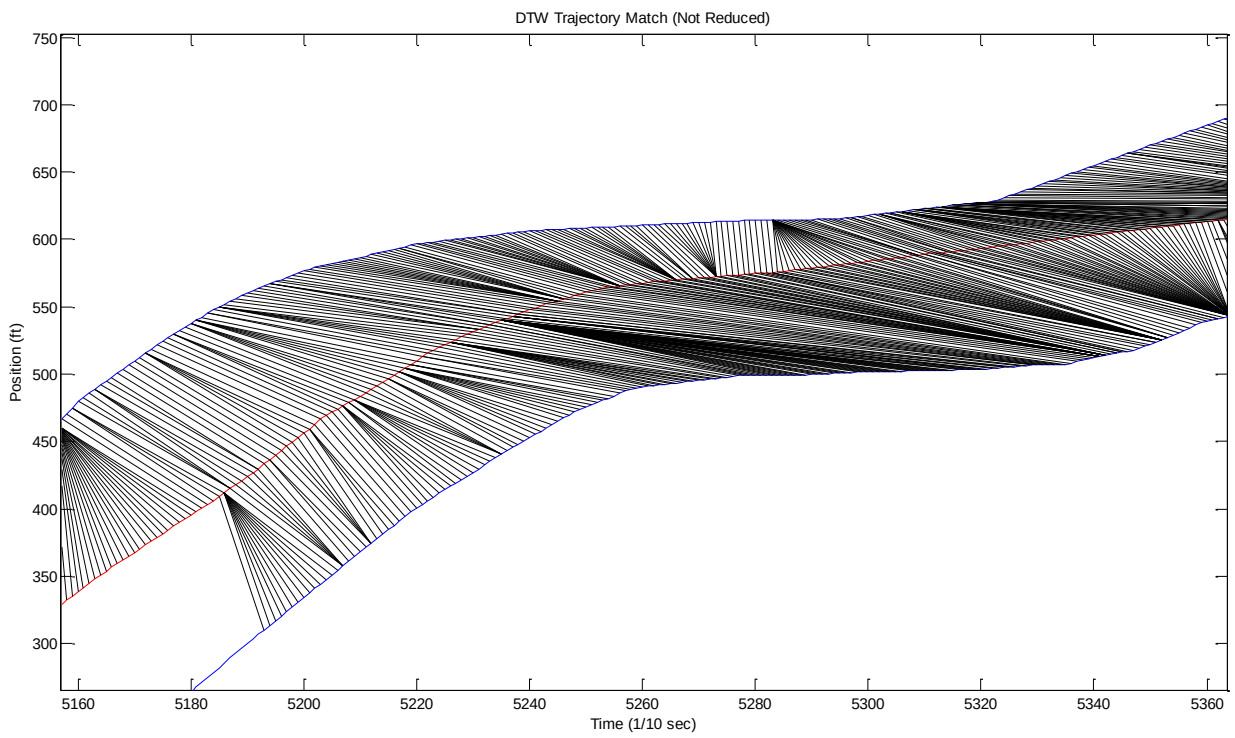


FIGURE 7 DTW trajectory match for unreduced data

5. LIMITATIONS AND CHALLENGES

DTW Input Data

Applying the DTW algorithm when working with vehicle trajectories requires some considerations for selecting input data, including the type of data, its time resolution, and the size of the datasets. The time-series input data for the DTW algorithm is two time-series datasets – one for the lead vehicle, and one for the following vehicle. Since the goal is to determine the driver's car-following parameters, the input data should come from the variable which forms the basis for the car-following model – velocity or acceleration. However, the algorithm is often applied with a distance measure used as the cost of aligning the datasets, where the distance is related to the difference in the two variables. This means that using velocity as the input will

match the leader's velocity to the follower's velocity as it changes with time. If acceleration is chosen as the input, the match is performed based on the response to the change in velocity. From a purely data analysis standpoint, if the time series data is smoothed to the point of being composed of nearly constant velocities, matching based on velocity will result in a large number of singularities, making the results very unrealistic.

Data resolution is another issue of concern when working with DTW for vehicle trajectory matching. High resolution (0.1 seconds) vehicle trajectory data is widely available, but significantly increases the computational resources required by the DTW algorithm, especially for large datasets. It may be desirable to reduce the datasets to only the most important data points for each time series. However, more dispersed data points may result in unrealistic or undesirable matches, and data reduction further reduces the number of points available for analysis. A multi-resolution approach may be necessary, where the matches are made between the reduced data points, followed by a second run through the algorithm for matching the trajectories between the reduced data points.

In many cases, the datasets may have different sizes, especially after any kind of data reduction algorithm is applied to the raw input data. The DTW algorithm can analyze datasets with different sizes, but this increases the number of singularities in the output data. While singularities may be undesirable in some cases, they may also be useful for different analyses, which will be discussed in the following section.

Singularities

Several variations of the DTW algorithm exist, each with their own unique features and components. Examples include Derivative DTW (*11*), Fast DTW (*12*), Multi-scale DTW (*13*), and DTW with Piecewise Aggregate Approximation or PDTW (*14*), among many others. Similarly, many modifications have been made to this algorithm to reduce the incidence of "singularities" – a case where a large section of one time series is matched with a single point in the other time series, sometimes in undesirable or unexpected combinations. For vehicle trajectory analysis, a singularity exists when the follower's reaction is mapped to multiple actions by the leader, or multiple actions by the follower are mapped to a single action by the leader. This also tends to occur in regions with constant velocity, and when a car-following parameter changes compared to that estimated in a previous time period. Viewed as part of a warp path in a matrix, as in Figure 3(D), singularities occur when the path moves vertically or horizontally, rather than diagonally. Horizontal and vertical steps in the warp path indicate changes in the reaction time τ (horizontal = increases, vertical = decreases, diagonal = same). A visual example of a singularity is provided in Figure 4 to better illustrate this phenomenon.

From a theoretical standpoint, singularities offer an interesting new perspective for analysis while simultaneously complicating that analysis. The match results for singularities imply a more complicated following behavior than the underlying model, where one stimulus could result in multiple responses, and vice versa. Additionally, singularities could also be used to classify drivers, where multiple responses to a single stimulus could indicate more aggressive behavior. However, to what degree singularities truly represent the leader-follower relationship, as opposed to artifacts of the algorithm, needs further study and analysis. Singularities have been considered undesirable in most studies using DTW, and a singularity must exist when datasets are not of equal size so that all points are matched. Additionally, a singularity may

present multiple solutions at one point for the time-dependent model parameters, which raises the issue of which value to use for calibration.

As a result of these issues, we cannot conclude that a singularity accurately reflects the leader-follower response. At the same time, we can only assume that a more complicated driver behavior is not present. We can implement some algorithm enhancements to reduce the presence of singularities, but care must also be taken to ensure that the methods used to reduce singularities also do not produce undesirable singularities which may also affect solution quality. For example, Figure 4 highlights a singularity which helps the algorithm transition from an impossibly-high *w* (nearly vertical slope) to more reasonable results.

Additional Enhancements for Singularity Reduction

Methods used to reduce the occurrence of singularities include, but are not limited to, windowing, slope weighting, and using different step patterns. Windowing is a process which limits the available number of matches to a single point based on a selected window width, which limits the size of a singularity. For vehicle trajectories, this method simply limits the calculated τ value for any given match to a range of reasonable values. This can also be used in conjunction with calculated d and *w* values for those match points to force the algorithm to always provide theoretically-acceptable matches.

Slope weighting adds coefficients to the cumulative cost terms in Eq. 6. Its implementation is a modified form of Eq. 6, which is shown in Eq. 9. The weight coefficients tend to encourage a more diagonal warp path through the cumulative cost matrix.

$$D(i,j) = C(i,j) + \min\big(D(i-1,j-1), \beta * D(i-1,j), \beta * D(i,j-1)\big) \tag{9}$$

As the coefficients increase, the warp path should become more diagonal in nature. A more diagonal warp path limits the presence of singularities, but may also have implications for the resulting model parameters from that warp path. Since the algorithm must produce matches near the beginning and end of the dataset, it may require some "warm-up time" before it produces reasonable results. This transition usually requires singularities so that the reaction time changes from zero to a reasonable solution. Thus, an attempt to limit the formation of singularities may extend that "warm-up time." Additionally, if the driver's behavior changes such that the model parameters are different at that location in the trajectory, a singularity should be expected, but large slope weights may disguise that change.

Different step patterns can also be implemented in the cumulative cost calculation. This requires changing Eq. 6 so that the algorithm works with cells in the cumulative cost matrix that are more than one step away in each direction. An example of this approach is given in Eq. 10 below.

$$D(i,j) = C(i,j) + \min\big(D(i-1,j-1), D(i-2,j-1), D(i-1,j-2)\big) \tag{10}$$

Again, this method increases the likelihood for a more diagonal warp path by forcing the path to move diagonally in addition to when it moves vertically or horizontally.

6. CONCLUSIONS & FUTURE STUDY

This paper describes a method for using the Dynamic Time Warping algorithm to calibrate an extension of Newell's car-following model incorporating time-dependent car-following parameters. The unique capabilities of the DTW algorithm may provide an efficient method for observing driver heterogeneity in car-following behavior, as well as the driver's heterogeneous situation-dependent behavior within a trip. Although the algorithm was made to analyze time-series data, several modification techniques are described to address specific challenges in this application and the algorithm solution quality for analyzing vehicle trajectories. A brief numerical experiment is presented with vehicle trajectory data extracted from the Next Generation Simulation (NGSIM) project, demonstrating the algorithm's ability to process large vehicle trajectory datasets, but significant data reduction and more algorithm modification may be necessary to produce more reasonable results. Additionally, singularities present an interesting match solution set to potentially help identify changing driver behavior, but they must be avoided to reduce analysis complexity and solution uncertainty. Further research will focus on algorithm enhancements, parameter validation methods, comparisons with alternative calibration methods, evaluating potential applications with other car-following models, and large-scale vehicle trajectory analysis to potentially explore situation-dependent driver behavior.

REFERENCES

1. Newell, G.F., 2002. A simplified car-following theory: a lower order model. Transportation Research Part B 36, 195–205.
2. Ossen, S. and S. Hoogendoorn, Car-Following Behavior Analysis from Microscopic Trajectory Data. *In Transportation Research Record: Journal of the Transportation Research Board*, No. 1934, Transportation Research Board of the National Academies, Washington, D.C., 2005, p. 13-21.
3. Ossen, S. and S. Hoogendoorn, Driver Heterogeneity in Car Following and Its Impact on Modeling Traffic Dynamics. *In Transportation Research Record: Journal of the Transportation Research Board*, No. 1999, Transportation Research Board of the National Academies, Washington, D.C., 2007, p. 95-103.
4. Ossen, S., & Hoogendoorn, S. P. Heterogeneity in car-following behavior: Theory and empirics. Transportation Research Part C: Emerging Technologies, *19*(2), 2011, p. 182-195.
5. Kim, J. & H. S. Mahmassani. Correlated Parameters in Driving Behavior Models: Car-Following Example and Implications for Traffic Microsimulation, *Transportation Research Board 90th Annual Meeting*, Washington, DC, January 23-27, 2011.
6. Laval, J.A., Leclercq, L., 2010. A mechanism to describe the formation and propagation of stop-and-go waves in congested freeway traffic. Philosophical Transactions of Royal Society A, 368(1928), 4519-4541.
7. Newell, G. Theories of instabilities in dense highway traffic, J. Op. Soc. Japan 5, 9 – 54 (1962)
8. Brockfeld, E., Kühne, R., Wagner, P.: Calibration and Validation of Microscopic Traffic Flow Models. Transportation Research Record: Journal of the Transportation Research Board, 1876, 62–70.
9. Ahn, S., Cassidy, M.J., Laval, J., 2004. Verification of a simplified car-following theory. Transportation Research Part B 38, 431–440.

10. Pavel S. "Dynamic Time Warping Algorithm Review ", Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA.
11. Keogh, E. & M. Pazzani. Derivative Dynamic Time Warping. In *Proc. of the First Intl. SIAM Intl. Conf. on Data Mining*, Chicago, Illinois, 2001.
12. Salvador, S., Chan, P., 2004. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. In: KDD Workshop on Mining Temporal and Sequential Data. Seattle, WA, USA, pp. 70–80.
13. Zinke, A. and Mayer, D. (2006). Iterative Multi Scale Dynamic Time Warping. Technical Report CG-2006-1, Universität Bonn.
14. Chu, S., E. Keogh, D. Hart & Michael Pazzani. Iterative Deepening Dynamic Time Warping for Time Series. In *Proc. of the Second SIAM Intl. Conf. on Data Mining*. Arlington, Virginia, 2002.
15. Next Generation Simulation (NGSIM), http://ngsim-community.org, Accessed June 20, 2011.