Driver Behavior Modeling Near Intersections Using Hidden Markov Model Based on Genetic Algorithm

Seifemichael B. Amsalu and Abdollah Homaifar
Electrical and Computer Engineering Department
North Carolina A&T State University
Greensboro, NC 27411 USA
e-mail: samsalu@aggies.ncat.edu, homaifar@ncat.edu

Abstract—**Driver behavior modeling plays a significant role in the development of Advanced Driver Assistance Systems (ADAS) for assisting drivers in different driving scenarios. One of the scenarios where high numbers of traffic accidents occur is road intersection. It is vital to develop driver behavior models near intersections in order for the ADAS to plan a proper action in avoiding accidents. In this paper, Hidden Markov Models (HMMs) for driver behavior near intersections are trained using Genetic Algorithm combined with Baum-Welch Algorithm based on the hybrid-state system (HSS) framework. HMM is usually trained using Baum-Welch which is easily trapped at local maxima. GA solves this problem by searching the entire solution space. Consequently, the best driver behavior model is trained. In the HSS framework, the vehicle dynamics are represented as a continuous-state system (CSS) and the decisions of the driver are represented as a discrete-state system (DSS). The continuous observations from the vehicle, such as acceleration, velocity and yaw-rate, are used by the proposed technique to estimate the driver's intention at each time step. The models are trained and tested using naturalistic driving data obtained from the Ohio State University, in an experiment with a sensor-equipped vehicle that was driven in the streets of Columbus, OH. The proposed framework improves the HMM accuracy in estimating the driver's intention when approaching an intersection with over 10% higher accuracy.**

Keywords-driver behavior modeling; hidden Markov models; genetic algorithm

I. INTRODUCTION

The development of autonomous vehicles leads to a mixed-traffic environment where human driven, semi-autonomous and autonomous vehicles must cooperate to assure a safe traffic flow. This requires estimation and prediction of the drivers' behaviors of these vehicles effectively in order to achieve road safety and safe driving. The techniques designed to estimate the intention of drivers are vital in the development of Advanced Driver Assistance Systems (ADAS) which results in automated driving [1]. The recent advancements in sensing, communication and computational technologies allow us to deploy the techniques designed for estimation of driver's intentions for active safety features such as antilock braking systems and adaptive cruise control to reduce road accidents [1].

The goal of this work is developing driver behavior models to make accurate estimation of the driver state during pre-crash scenarios. The development of multi-agent models is based on a framework developed in Ohio State University (OSU) [2], [3]. The framework takes dynamic inputs from the changing environment and other behaviors, and simulates the perception, attention, cognition, and control behavior of the driver by applying different mathematical or symbolic methods [3], [4]. In this paper, an algorithm that estimates the driver's intention near a road intersection controlled by a traffic signal is proposed and validated using naturalistic driving data. Estimating from a set of observations whether the driver will stop, turn right, turn left, or go straight safely according to traffic signal indicator near the intersection is the objective. Different behaviors of the drivers results in variations in observations which must be taken into account in the classification process. For example, when a driver decides to "Turn Left" at an Intersection, the turn left signal blink, the speed reduces and brake light illuminate, and he/she turns left finally. The driver behavior estimation is the task of observing the continuous vehicle dynamics and estimating the driver decisions that result in those observations. Here, the combination of vehicle and driver is referred as "driver" and the trajectory shown by the combination is referred as "driver behavior" [5]. The overall work is based on the Hybrid-State System (HSS) framework [6].

In this study, the Hidden Markov Model (HMM) with Genetic Algorithm (GA) optimization is used to estimate the driver's decisions that are the discrete-state in the HSS framework, as the vehicle approaches an intersection. Usually, Baum-Welch algorithm is used to train HMM with random initial guess of the parameters and then more accurate parameters are computed at each iteration until it converges. Baum-Welch is easily trapped to local maxima which are near the initial values and doesn't converge to the global optimum that gives best HMM model [7]. Consequently, an algorithm that solves the local maxima problem and searches the entire solution space is needed. Genetic Algorithm is an evolutionary optimization technique that imitates the evolution processes in nature [8], [9]. It searches the global optima from the solution space based on multiple points simultaneously using crossover and mutation operators. In order to make faster the convergence of the HMM training with GA only, Baum-Welch is applied between generations of GA. The HMM-GA technique has been widely utilized in many applications including biological sequence analysis, handwritten recognition, human motion pattern classification, speech recognition, etc. [10], [11], [12], [13] and its outperforms Baum-Welch only training. Overall, the HMM-GA is combined with the HSS framework to achieve driver behavior estimation. The HMM-GA is utilized as a mathematical technique that relates the discrete-states (driver decisions) and continuous-states (continuous observations) of the HSS framework. In this paper, the term HSS+HMM-GA refers to this combination. The performance of the proposed HSS+HMM-GA method for driver behavior estimations is compared with HSS + Hidden Markov Model (HMM) reported in [5], [14]. The accuracy percentage of the HMM-GA model exceeds the HMM in estimating the driver's decisions near intersections.

Many researchers have studied the problem of driver behavior modeling including near a road intersection scenario. In [15], the authors developed graphical models, HMMs and their extensions for different driver maneuvers with a focus on the context effect on the driver's performance. The contextual traffic information is used to predict the maneuvers with the real-time vehicular observations. In [16], a framework for a cognitive model of human behavior is proposed with a HMM method as the driver behavior recognition for emergency and normal lane changes. The authors trained the HMM models using driver behavior data from a driving simulator. In [17], a model for recognition of driving events using discrete HMMs is presented utilizing longitudinal and lateral acceleration and speed data from a real vehicle in a normal driving environment. In [14], HMM and SVM approaches for driver behavior modeling near a road intersection using naturalistic driving data are designed and their performance is compared.

In this paper, the HMM based on Genetic Algorithm (HMM-GA) method for a driver behavior estimation from observable parameters of the vehicle using the HSS framework is presented. An example that describes the problem of driver estimation is presented in section II. A brief discussion on HSS, HMM, and Hybrid HMM training with Genetic and Baum-Welch Algorithms (GA-BW) is provided. The GA evolutionary training of the HMM parameters is explained in detail here. The data collection and data analysis is described in section III. Section IV discusses the results of the proposed technique. The results are also compared with the HMM model trained only with Baum-Welch algorithm. Finally, the conclusion and future work of the paper is presented.

II. DRIVER BEHAVIOR ESTIMATION FRAMEWORK

In section I, the driver behavior estimation is the task of estimating the driver's decisions from observable continuous vehicle dynamics. Fig. 1 shows an example intersection scenario. When the autonomous red car approaches the intersection, it tends to turn left as shown by the dashed red line. Before determining its maneuver, on-board sensors are used to alert the vehicles of another human-driven vehicle with the right of way, as shown in green. The red car, before turning left, must determine which path the green car will follow. Here, if the green car is turning right, the red car can turn left; otherwise it must stop and let the green one pass. Hence, the red car needs to determine the intention of the green car's driver from observations obtained through the use of on-board sensors such as GPS, lidar, and radar or vehicle-to-vehicle (V2V) communication [5].

A. Hybrid State System (HSS) Framework

A Hybrid-State System (HSS) have been modeling the hierarchical interaction of a discrete-state system and a continuous-state plant in various applications including autonomous vehicles [18] and HSS estimation [6]. The behavior of a vehicle and its driver can be tracked, estimated and predicted by HSS model that represents the interaction of the vehicle and the driver. The HSS setting that consists of a higher level discrete-state system (DSS) part and a lower level continuous-state system (CSS) part is shown Fig. 2. According to discrete events a driver reacts and makes corresponding decisions on the higher level, and based on the driver decisions the vehicle follows a specified continuous trajectory. The HSS system formulation and equation is developed in [19].

B. Hidden Markov Models (HMM)

The relationship between the CSS and DSS parts of HSS discussed above can be mathematically modeled using HMMs [20], [21] that provides a stochastic relationship between the vehicle dynamics and driver state. Here, the changes in CSS vehicle dynamics lead to changes in the driver states which are not predefined and determined by training observation data using the model.

A HMM is a stochastic model of the dynamic process of two related random processes. A set of stochastic processes that produces the sequence of observed symbols are used to observe an underlying stochastic process that is not observable (hidden states). A HMM consists of a set of N distinct "hidden" states of the Markov process S = $\{s_1, s_2, ? s_N\}$ and a set of M observable symbols per state V=$\{v_1, v_2, ? v_M\}$. The overall HMM model is defined as follows (with q_t and o_t denoting the state and observation symbol at time t respectively).

The HMM is specified by a set of parameters (A, B, π):

- The prior probability distribution $\pi = \{\pi_i\}$ where $\pi_i = P(q_1 = s_i)$ are the probabilities of s_i being the first state in a state sequence.
- The transition probability matrix $A = \{a_{ij}\}$ where $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$ are the probabilities to go from state s_i to state s_j.
- The emission (observation) probability matrix $B = \{b_{ik}\}$ where $b_{ik} = P(o_t = v_k \mid q_t = s_i)$ are the probabilities to observe v_k if the current state is s_i.

The above emission probability matrix is defined for discrete HMM case. In our problem, the observation symbols are continuous vehicle dynamics (acceleration, velocity and yaw-rate), so the HMM development is based on probability density functions (pdfs). The emission pdfs for a continuous valued observation is described by a set of probability density functions (PDFs), $b_i(o_t) = p(o_t \mid q_t = s_i)$, over the observation space for the system being in state s_i. In this paper, the most general emission distribution, that is modeled as a joint Gaussian distribution $\mathrm{N}(\mu, \Sigma)$, called Gaussian Mixture Model (GMM) is used, with

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1M}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{M1}^2 & \sigma_{M2}^2 & \cdots & \sigma_{MM}^2 \end{bmatrix}. \quad (1)$$

The GMM is of the form

$$b_i(o_t) = p(o_t \mid q_t = s_i) = \sum_{m=1}^{M} c_{im} \mathrm{N}(\mathrm{o}_t; \mu_{im}, \Sigma_{im}) \quad (2)$$

where c_{im} stands for the mixture coefficient for the m^{th} mixture in the i^{th} state and $1 \le i \le N$. The N stands for the pdf of a normal distribution with mean μ and covariance Σ measured at observation o_t. Mixture coefficient c_{im} satisfies the following constraint:

$$\sum_{m=1}^{M} c_{im} = 1 \quad 1 \le i \le N, 1 \le m \le M \quad (3)$$

Such that the pdf is normalized to be

$$\int_{-\infty}^{+\infty} b_i(o) do = 1, \quad 1 \le i \le N. \quad (4)$$

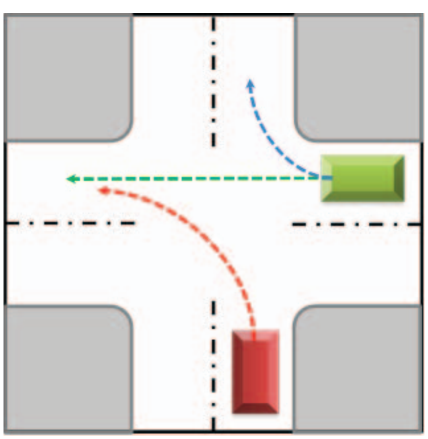

Figure 1. A simple intersection senario. Given the right of way is for the green car, for the red car to turn left, it must determine if the green car is going straight or turning right.

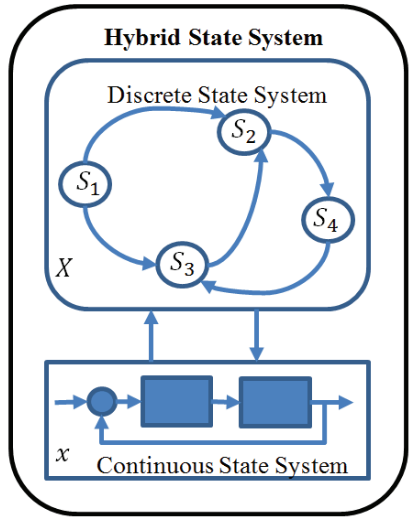

Figure 2. Hybrid state system setting.

With the above definitions, an HMM can be completely specified by the pdf of a normal distribution N, and the five probability measures $\lambda = (A, \pi, c, \mu, \Sigma)$.

To infer the discrete-state of the driver from the continuous observations of the vehicle, HMM models are trained using an iterative procedure called the *Baum-Welch* method (Expectation Maximization) [22], [23]. It computes the maximum-likelihood estimates of the HMM model parameters $\lambda = (A, \pi, c, \mu, \Sigma)$ using the *forward* and *backward algorithm* [23]. Once the models, $\lambda_1, \lambda_2, \ldots, \lambda_n$, are trained, the *forward algorithm* is used to determine the driver states. For a given sequence of observations, $O = \{o_1, o_2, \ldots, o_t\}$, the forward algorithm computes the probabilities $P(O \mid \lambda_i), i = 1, 2, \ldots, n$ for each model. The model λ_i with the highest likelihood probability at a given time step represents the estimated driver state i. Thus, the driver's state at time t is determined as

$$S(t) = \arg\max_i P(o_1, o_2, \ldots, o_t \mid \lambda_i) \quad i = 1, \ldots, n. \quad (5)$$

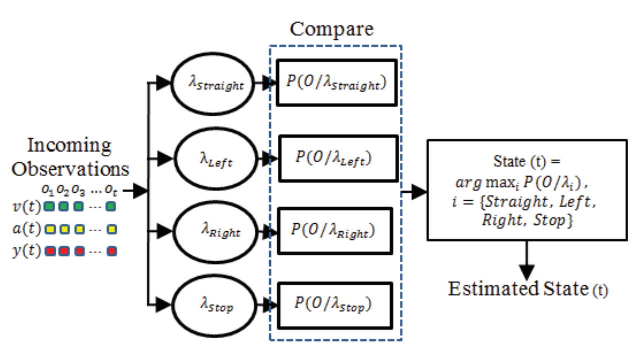

Figure 3. Estimation of the state that best describes the incoming observations sequences.

The general idea of this technique is shown in Fig. 3. The final compare step represents (5). Since the focus of this paper is in vehicle actions near a road intersection, models λ_i : straight, left, right and stop are the possible driver decisions near the intersection. Genetic and Baum-Welch Algorithms are used to train the HMM models in this paper and discussed in the following subsection.

C. *Hybrid HMM Training with Genetic and Baum-Welch Algorithms (GA-BW)*

The Genetic Algorithm (GA) is used to train HMM model that gives a maximum likelihood probability for the given training data. As a result, the error of the model will be minimum in estimating the driver state at time t . The parameters of the HMM model are represented as genes of a chromosome using real valued GA. A random population consisting of several candidate solutions is generated and based on the survival fitness, the population transforms into fit individuals after some number of generations. To apply the GA, a genetic representation to the problem and fitness function to evaluate the solution domain should be defined. Due to slow convergence and high computational power needed in GA, applying Baum-Welch algorithm on every individual in the GA population allows a faster convergence in training the HMM model. For every 20 generations in GA, 10 iterations of Baum-Welch are done in this work. In the following sub-subsections, the detailed explanation of the GA used is presented.

1) Chromosome Representation: In GA, it is necessary to have a chromosome representation of a solution (HMM model) to solve the optimization problem. The chromosome is composed of a set of basic elements called genes. In this work, the chromosome is composed of every element of the probability matrices of the model A, B and π , as the the observations are continuous B is the same as μ and Σ . The chromosome is represented by concatenating the rows of each matrix in the HMM model parameters $\lambda = (A, \pi, \mu, \Sigma)$. Thus, the model is encoded into a string of real numbers as shown in Fig. 4.

a_{11}	a_{12}	…	a_{1N}	a_{21}	…	a_{2N}	…	a_{NN}	μ_{11}	μ_{12}	…	μ_{1M}	μ_{21}	…	μ_{2M}	…	μ_{NM}

Σ_{11}	Σ_{12}	…	Σ_{1M}	Σ_{21}	…	Σ_{2M}	…	Σ_{MM}	π_1	π_2	…	π_N

Figure 4. Chromosome representation.

2) Fitness Function: The fitness function evaluates the quality of a solution and is defined over the above genetic representation. The higher fitness value reflects the chance of the chromosome to be chosen for the next generation. In this work, the log likelihood [7] is used to define the fitness of an individual solution (model). It represents the average probability P_n that the training observation sequences $O_1, O_2, \ldots, O_K$ have been generated by the current model parameters λ_n . The fitness function is defined as follows:

$$P_n = (\sum_{k=1}^{K} \log(P(O_k \mid \lambda_n))) / K . \quad (6)$$

where $P(O_k \mid \lambda_n)$ is calculated by the forward algorithm presented in [23].

The genetic representation and the fitness function are defined to come up with the best HMM model that represents well the driver decisions near an intersection. The GA then proceeds to initialize a population of solutions and improves it through repetitive application of selection, crossover, and mutation operations until some number of generations is reached.

3) Initialization (Population Generation) and Fitness Evaluation: The initialization step in most GA applications should be made as simple as possible and generated in random. Here, the initial population (HMM models) is generated by Baum-Welch algorithm with one iteration. After the initial population is generated, the fittest value of each individual (chromosome) is evaluated. The fitness function that is defined in (6) is used. A population size of *P=20* individuals are used in this work.

4) Selection: The selection process determines which of the chromosomes from the current population will mate to create new chromosomes. There are several selection methods. In this work, *tournament selection* method with tournament size *k=3* is applied [8] . In this method, k individuals are selected randomly from the population and the individual with the maximum fitness value from the group will be part of the new population in a mating pool for reproduction using crossover and mutation to generate the next generation of individuals. This results in a population that contains individuals with best fitness value.

5) Reproduction: Crossover and Mutation: The selected parents in the mating pool mate one-another to produce new offspring through crossover. Crossover is made on two parents by selecting one of the parents sequentially from the mating pool and the other-one randomly. The two parents are crossed using simple arithmetic crossover at a crossover point j as follows [24]:

Parents:

$$P_1 = \{x_1, x_2, \ldots, x_n\} \text{ and } P_2 = \{y_1, y_2, \ldots, y_n\}$$

Offspring:

$$C_1 = \{x_1 \ldots, x_j, \beta * (x_{j+1} \ldots, x_n) + (1-\beta) * (y_{j+1} \ldots, y_n)\}$$

And

$$C_2 = \{y_1 \ldots, y_j, \beta * (y_{j+1} \ldots, y_n) + (1-\beta) * (x_{j+1} \ldots, x_n)\}$$

where β is a random number in the range [0, 1]. In this work, a constant value $\beta = 0.5$ and a crossover rate of $p_c = 0.8$ are used. The children compete with the parents to be included in the next generation with *deterministic* *crowding* [25]. This increases the diversity of the population and the quality of the solutions gets better. Each child (C_i) replaces the nearest parent if it has higher fitness. P_1 against C_1 and P_2 against C_2 or P_1 against C_2 and P_2 against C_1 as shown in Fig. 5 where $f(.)$ is the fitness function and $d(.,.)$ is an Euclidean distance measure between the two chromosomes.

After the crossover operator is applied on the population in the mating pool, the new population undergo through mutation operation. Mutation is done on every individual in the population with a probability of $pm_1 = 0.001$, $pm_2 = 0.006$, $pm_3 = 0.005$ and $pm_4 = 0.004$ for the initial distribution π_i, transition probabilities a_{ij}, mean μ_{ij} and co-variance Σ_{ij} parts of the chromosome used to represent HMM model respectively. When a gene x_j is mutated, its value will be changed to a new real value $x_j^* = rand(1) * x_j + x_j$ where $rand(1)$ generates a uniformly distributed random number in the range [0, 1]. As a result, new genetic patterns will be introduced to the new generation. After mutation the chromosome is normalized to satisfy the HMM properties: $\sum_{i=1}^{N} \pi_i = 1$, $\sum_{j=1}^{N} a_{ij} = 1$ where $1 \le i \le N$, and the co-variance matrix Σ should be positive definite matrix.

6) Termination: In this work, the GA runs until the maximum number of generations, *G=200*, is reached. The number can be increased to get better chromosomes. But as the maximum number of generations increases, the computational time of the algorithm increases.

After the GA is completed, the chromosome that is the best from the last generation, i.e. the individual with the maximum fitness value, is taken to be the best HMM model. To determine the drivers state at time t, this chromosome provides the HMM model λ_i with the highest likelihood probability at a given time step, representing the estimated driver state for a sequence of observations, $O = \{o_1, o_2, \ldots, o_t\}$,. Thus, the error in estimating the driver state at time t decreases as compared to training the HMM model with only Baum-Welch algorithm.

III. Data Collection and Analysis

In this study, the driver behavior information data set collected by the Ohio State University researchers [5] is used to train the mathematical models discussed in section II. The focus of this work is estimating the states that represent the high-level behavior of the driver from the observable parameters. It is assumed that we are able to collect the low-level continuous observations such as velocity, yaw-rate, and acceleration, given the correct combination of sensors.

If $(d(P_i, C_i) + d(P_j, C_j)) \le (d(P_i, C_j) + d(P_j, C_i))$ **then**
- **If** $f(C_i)$ is better than $f(P_i)$ **then** replace P_i with C_i.
- **If** $f(C_j)$ is better than $f(P_j)$ **then** replace P_j with C_j.

Else
- **If** $f(C_i)$ is better than $f(P_j)$ **then** replace P_j with C_i.
- **If** $f(C_j)$ is better than $f(P_i)$ **then** replace P_i with C_j.

Figure 5. Pseudocode algorithm for deterministic crowding [25].

A 2012 Honda Accord car equipped with sensors [5] is used to collect the data. The sensors are: *NovAtel GPS unit*- it provides timestamp of reading, GPS longitude, GPS latitude and others; *Honda Accord controller area network (CAN) bus*- it provides timestamp of reading, lateral acceleration, speed, yaw rate, steering wheel angle, odometer, turn signals and others. *Three HD cameras*- they provide views of the front, right side and left side of the vehicle.

The vehicle is driven through selected paths that represent a typical road situations encountered in daily driving tasks by selected participant drivers. The driving scenarios of interest are road intersections (Straight, Left turn, Right turn and Stop) and highway driving (merging and entering/exiting ramps).

The videos from collected data are manually marked to extract the ground truth data representing each driving scenario. For instance, in the video when a vehicle approaching an intersection and turn right, the time span was marked as "Right Turn" and the corresponding time series observations of velocity, yaw-rate and acceleration were extracted within this time span. Examples of 11 seconds time series observations are shown on Fig. 6 for different maneuvers. These observations are enough to accurately describe an intersection driving behavior as shown in [5], [26]. The extracted time-series observations were used to train the models. The HMM models trained using hybrid GA-BW algorithms and Baum-Welch algorithm only as shown in Fig. 3 are tested and compared on the accuracy of estimating the states at each time step.

IV. Results

In this work, the HMM based on Genetic Algorithm (HMM-GA) is applied to model the driver behavior at an intersection scenario and compared with the HMM based model. The time series observations of a vehicle dynamics at time $t = \tau$ are used to train the models. The HMM model represents the vehicle trajectory from the time series observations for classifying the vehicle maneuver. A Confusion Matrix, where each row represents the number of instances in an actual class and each column represents the number of instances in a predicted class, is used to evaluate the performance of the models.

The performance of the HMM-GA model is shown in the confusion matrix on Table-I for 70% training and 30% testing set at $t = \tau$ with training parameters discussed in Section II. The model is trained with $M = 3$ GMMs for the velocity, yaw-rate, and acceleration observations and $N = 4$ possible hidden states. Here, the time series data for 6 Left Turn, 18 Straight, 5 Right Turn and 21 Stop maneuvers converted to 655 Left Turn, 1964 Straight, 545 Right Turn and 2279 Stop time series observation vectors at a given time $t = \tau$ respectively. The conversion is done at time step of 0.1 seconds for a maneuver length of 11 seconds that is 6 seconds before and 5 seconds after entering the intersection. This makes the total number of time series vectors in the data set to be 5443 with corresponding labels that is divided into 70% training and 30% testing set.

TABLE I. CONFUSION MATRIX FOR HMM-GA USING 30% TESTING SET AT $t = \tau$

Actual Maneuvers at $t = \tau$	**Predicted Maneuvers at $t = \tau$**			
	Straight	***Left Turn***	***Right Turn***	***Stop***
Straight	440	25	43	39
Left Turn	0	272	40	15
Right Turn	0	77	245	5
Stop	4	100	91	459

Accuracy = 76.33%

The proposed method is compared with its HMM counterpart as shown in the following confusion matrix using the 30% testing set and also reported in [14]. The same as above the model is trained with $M = 3$ GMMs and $N = 4$ possible hidden states using the modified version of [27]. Table II shows the confusion matrix based on the HMM model. In estimating the states at each time step, the accuracy of the HMM-GA model is better than the HMM model trained with Baum-Welch only as shown in Table-I. The Genetic Algorithm (GA) gives the capability to search the entire solution space to maximize the log-likelihood of the HMM model. As a result the HMM-GA model gives the best recognition performance over the HMM model.

The estimated states for four individual example maneuvers: straight, left turn, right turn and stop observation sequences at an intersection, using the HMM and the HMM-GA methods discussed above are shown in Fig. 7. This figure describes the estimators output at a given time $t = \tau$ for HMM-GA and HMM. For instance, Fig. 7 (a) shows the state estimation of an example "straight maneuver sequence" with ground truth corresponding to a vehicle going straight through an intersection using the developed models, the HMM and the HMM-GA models. At every time step (5) is evaluated for the HMM method as shown in Fig. 3 and the HMM that best describes the sequence until that point is the estimated state at a given time $t = \tau$. The same way, at every time step, the state estimation for the sequence at $t = \tau$ using the HMM-GA method is done by evaluating (5) and also as shown in Fig. 3. This is similarly repeated for sample maneuvers observation sequence with ground truth corresponding to a vehicle approaching an intersection and turning left, right and stopping. The figures show that overall the HMM-GA method outperforms the HMM one.

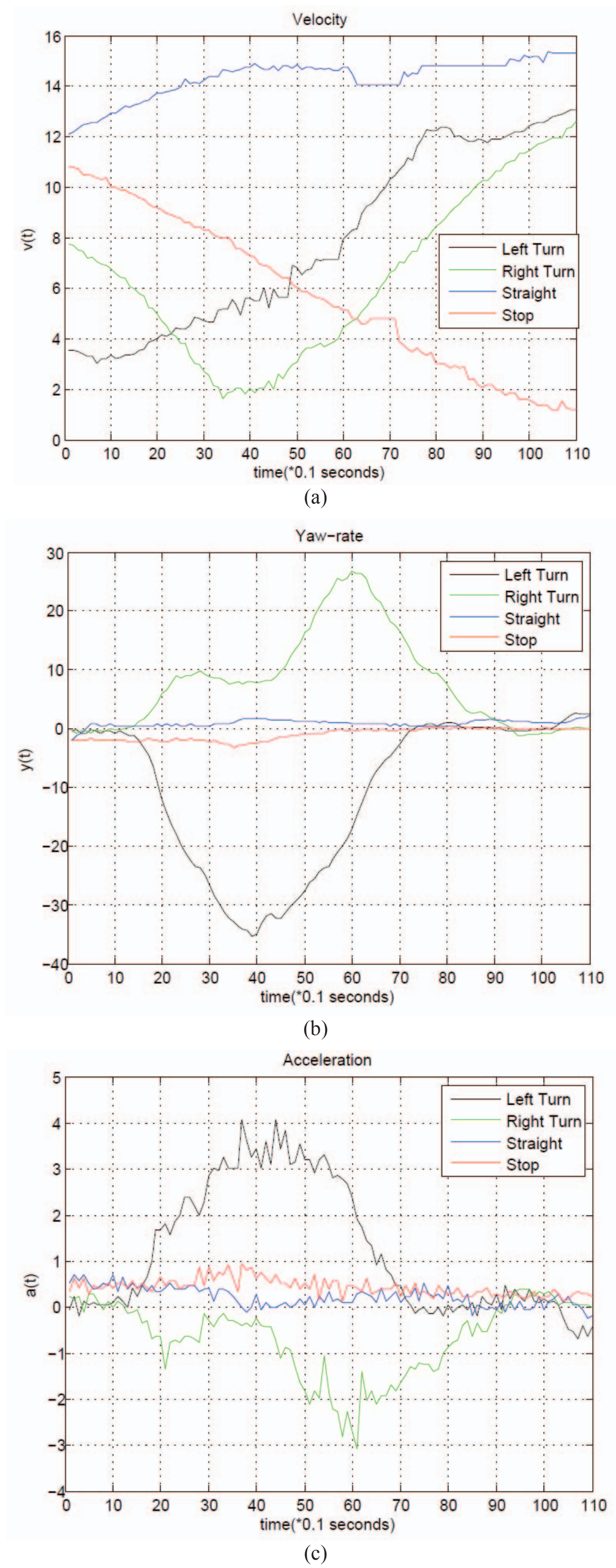

Figure 6. Time series observations for 11 seconds (a) Velocity (b) Yaw-rate (c) Acceleration.

TABLE II. CONFUSION MATRIX FOR HMM USING 30% TESTING SET AT $t = \tau$

Actual Maneuvers at $t = \tau$	Predicted Maneuvers at $t = \tau$			
	Straight	***Left Turn***	***Right Turn***	***Stop***
Straight	361	46	111	136
Left Turn	4	174	38	2
Right Turn	16	1	142	22
Stop	95	103	114	447

Accuracy = 62.03%

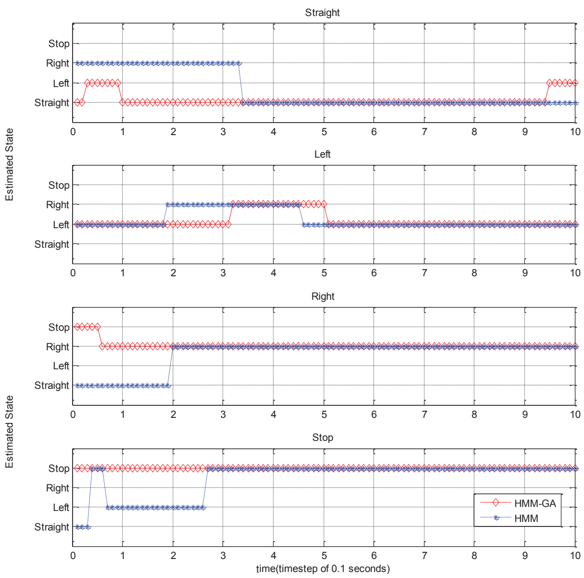

Figure 7. Estimation of states at each timestep using HMM-GA and HMM methods for four different individual vehicle maneuvers (a) Straight, (b) Left, (c) Right (d) Stop.

V. CONCLUSION AND FUTURE WORK

This work has presented an HMM-GA method based on the HSS framework to model a driver behavior near road intersections. It uses GA to optimize the HMM model parameters to come up with a model with best recognition rate. It combines the HMM-GA with the HSS framework to integrate the discrete driver state with the continuous vehicle dynamics. It gives accurate results with a performance comparable to human observer. The HMM model represents the stochastic process that produces the observations with the intuition of Markov chains. The proposed method is also compared with the HMM, that is trained with Baum-Welch algorithm only, based on HSS in estimating the state of the driver at an intersection. The HMM-GA outperforms HMM in generalizing the pattern recognition problems as a result of global optimization. The HMM-GA has achieved over 10% improvement on the HMM in driver behavior estimation near an intersection. In future work, the HMM model sensitivity on parameter variation that arise due to noise in the training observations will be done. Moreover, a comprehensive solution will be achieved if the driver's unclear decision near the intersection is considered in the estimation technique. In addition, the proposed method will also be applied in different driving scenarios of interest including lane change, merging and entering/exiting ramps and other near crash events using large scale dataset from SHRP2.

ACKNOWLEDGMENT

This research is supported by the US Department of Transportation (USDOT) under University Transportation Center (UTC) Program (DTRT13-G-UTC47). The second author would like to acknowledge the support from Air Force Research Laboratory and OSD for sponsoring this research under agreement number FA8750-15-2-0116. And also, authors would like to thank Ümit Özgüner (PhD) and Arda Kurt (PhD) from Ohio State University for helpful discussions.

REFERENCES

[1] W. D. Jones, "Keeping cars from crashing," IEEE Spectrum, vol. 38, no. 9, pp. 40-45, 2001.

[2] A. Kurt, and U. Özgüner, "A probabilistic model of a set of driving decisions," In 14th International IEEE Conference Intelligent Transportation Systems (ITSC), pp. 570-575, 2011.

[3] U. Özgüner, T. Acarman, and K. Redmill, Autonomous ground vehicles, Artech House Publishers, 2011.

[4] B. Song, and D. Delorme, "Human driver model for smart AHS based on cognitive and control approaches," In ITS America conference, 2000.

[5] V. Gadepally, A. Krishnamurthy, and U. Özgüner, "A Framework for Estimating Driver Decisions near Intersections," Trans. Intell. Transp. Syst., vol. 15, no. 2, pp.637-646, 2014.

[6] A. Kurt, J. Yester, Y. Mochizuki, and U. Özgüner, "Hybrid-state driver/vehicle modelling, estimation and prediction," In Proceedings 13th IEEE ITSC, pp. 806-811, 2010.

[7] M. Oudelh, and R.N Ainon, "HMM parameters estimation using hybrid Baum-Welch genetic algorithm," Information Technology (ITSim), Int. Symp. Vol. 2 , 2010.

[8] D. E Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[9] A. Homaifar, S. H.-Y. Lai, and X. Qi, "Constrained optimization via genetic algorithms," Simulation, vol. 62, no. 4, pp.242 -254, 1994.

[10] X. Zhang, Y. Wang, and Z. Zhao, "A Hybrid Speech Recognition Training Method for HMM Based on Genetic Algorithm and Baum Welch Algorithm," Innovative Computing, Information and Control, 2007. ICICIC '07.

[11] Shing-Tai Pan and Tzung-Pei Hong, "Robust Speech Recognition by DHMM with A Codebook Trained by Genetic Algorithm," Journal of Information Hiding and Multimedia Signal Processing, vol.3, no.4, Oct. 2012.

[12] T. Bhowmik, S. Parui, M. Kar, and U. Roy, "HMM Parameter Estimation with Genetic Algorithm for Handwritten Word Recognition," Volume 4815, Lecture Notes in Computer Science, pp 536-544, Springer Berlin Heidelberg, 2007.

[13] S. Manabe, T. Hatanaka, K. Uosaki, N. Tabuchi, T. Matsuo, K. Hashizume, "Training Hidden Markov Model Structure with Genetic Algorithm for Human Motion Pattern Classification," SICE-ICASE, International Joint Conference, 2006.

[14] S. Amsalu, A. Homaifar, F. Afghah, S. Ramyar, and A. Kurt, "Driver Behavior Modeling near Intersections Using Support Vector Machines based on Statistical Feature Extraction," in IEEE Intelligent Vehicles Symposium, June 28 - July 1, 2015.

[15] N. Oliver and A. Pentland, "Graphical models for driver behavior recognition in a smart car," in Proc. IEEE IV Symp., 2000, pp. 7-12.

[16] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, "A driver behavior recognition method based on a driver model framework," SAE Trans., vol. 109, no. 6, pp. 469-476, 2000.

[17] D. Mitrovic, "Reliable method for driving events recognition," IEEE Trans. Intell. Transp. Syst., vol. 6, no. 2, pp. 198-205, Jun. 2005.

[18] A. Kurt and U. Özgüner, "Hybrid state system development for autonomous vehicle control in urban scenarios," in Proceedings of the IFAC 2008 World Congress, pp. 9540-9545, 2008.

[19] M. Dogruel and U. Özgüner, "Discrete and hybrid state system modeling and analysis," Turkish Journal of Electrical Engineering and Computer, vol. 5, no. 2, pp. 263-286, 1997.

[20] L. Rabiner and B. Juang, "An introduction to hidden Markov models," IEEE ASSP Mag., vol. 3, no. 1, pp. 4-16, Jan. 1986.

[21] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989.

[22] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," Ann. Math. Stat., vol. 37, no. 6, pp. 1554-1563, Dec. 1966.

[23] J. Bilmes,"A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Int. Comput. Sci. Inst., BerkeleSewumehony, CA, Tech. Rep. ICSI-TR- 97-021, 1997.

[24] S. Picek, D. Jakobovic and M. Golub, "On the Recombination Operator in the Real-Coded Genetic Algorithms," Proceedings of the IEEE Congress on Evolutionary Computation, pp.3103 -3110, Jun. 2013.

[25] M. Lozano, F. Herrera, and J. R. Cano, "Replacement strategies to preserve useful diversity in steady-state genetic algorithms," Information Sciences, vol. 178, Issue 23, pp 4421-4433, Dec. 2008.

[26] G. Aoude, V. Desaraju, L. Stephens and J. How, "Behavior Classification Algorithms at Intersections and Validation using Naturalistic Data," IEEE Intell. Veh. Sympo, 2011.

[27] K. Murphy, Hidden Markov Model (HMM) toolbox for Matlab. [Online]. Available: http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html.