2011 14th International IEEE Conference on
Intelligent Transportation Systems
Washington, DC, USA. October 5-7, 2011

# Driver/Vehicle State Estimation and Detection

Vijay Gadepally, Arda Kurt, Ashok Krishnamurthy, and Ümit Özgüner

{gadepalv, kurta, akk, umit}@ece.osu.edu
Electrical and Computer Engineering,
The Ohio State University, Columbus, OH

*Abstract*— **The authors present a cyber-physical systems related study on the estimation and prediction of driver states in autonomous vehicles. The first part of this study extends on a previously developed general architecture for estimation and prediction of hybrid-state systems. The extended system utilizes the hybrid characteristics of decision-behavior coupling of many systems such as the driver and the vehicle; uses Kalman Filter estimates of observable parameters to track the instantaneous discrete state, and predicts the most likely outcome. Prediction of the likely driver state outcome depends on the higher level discrete model and the observed behavior of the continuous subsystem. Two approaches to estimate the discrete driver state from filtered continuous observations are presented: rule based estimation, and Hidden Markov Model (HMM) based estimation. Extensions to a prediction application is described through the use of Hierarchical Hidden Markov Models (HHMMs). The proposed method is suitable for scenarios that involve unknown decisions of other individuals, such as lane changes or intersection precedence/access. An HMM implementation for multiple tasks of a single vehicle at an intersection is presented along with preliminary results.**

## I. Introduction

The Cyber-Physical Systems (CPS) Group at The Ohio State University is active in a project that addresses *autonomous vehicles operating safely in mixed-traffic urban environments*. An autonomous vehicle must be capable of interacting with humans, other vehicles (both autonomous and non-autonomous), external physical effects, and internal and external software modules. In the interaction between autonomous and non-autonomous, or human driven vehicles, the task of successful detection and prediction of other vehicles and their future behavior are essential for autonomous vehicles to recognize, avoid and/or mitigate the threat of traffic accidents. One area of particular area of interest is near intersections, where vehicles approaching an intersection must be able to understand the intention of other vehicles. While this study focuses on the case of intersection approach, the developed models can be generalized for other situations of interest.

A first step for developing autonomous vehicles and collision avoidance systems is the detection or estimation of driver behavior of other vehicles, or the "driver state". The term "driver" is used to describe to combination of human operator and vehicle. Thus, estimation of driver behavior involves estimation of target vehicle state, along with prediction of future vehicle (and driver) paths.

A hybrid-state system (HSS) representation [1] of the driver-vehicle coupling, which has been useful for a number of automotive-related research areas, such as autonomous vehicles, is utilized in this study for analysis, modeling and estimation of the driver/vehicle behavior. The novelty of the proposed algorithm, among other hybrid-state estimation, research is that the direction of estimation propagation is from the vehicle states to the driver states, exploiting the uniformity of the vehicle model under various situations and borrowing tools from communication-based discrete-state estimation. Real-time discrete state estimation is carried out either through simple internal models (such as a rule based method which has been used in previous work [1]) or through Hidden Markov Models (HMMs) [2]. HMMs are doubly stochastic tools that can help identify underlying relationships between observable and hidden states. HMMs have had success in topics such as speech recognition [3], and driving events [4]. Driver/vehicle behavior prediction is also discussed through the proposed development of Hierarchical HMMs (HHMMs) conjoined with the HSS. HHMMs [5] are proposed as a tool to formalize the relationship between states of the DSS.

The aim of this paper is to discuss suitable models for estimating driver state from low-level CSS parameters. In Section II, we provide an overview of the HSS. Sections III, and IV discuss the Driver Estimation architecture along with preliminary results. A discussion of the prediction aspects of the system are given in Section V, and we conclude in Section VI.

## II. Hybrid-State Systems, and System Layout

In order to estimate, track and predict the behavior of the vehicle and its driver, the interaction between the vehicle and the driver is captured in a Hybrid-State System (HSS) model, which consists of a discrete-state system (DSS) higher level and a continuous-state system (CSS) lower level, as seen in left part of Figure 1. The driver reacts to discrete events, makes corresponding decisions on the higher level, and the vehicle follows continuous trajectories according to driver intention. This coupling of systems with different domain characteristics has been modeled as a HSS for a variety of applications, including hybrid-state controllers for autonomous vehicles [6].

The interaction between the modes or the states in the DSS is modeled as a Finite State Machine (FSM), with discrete state, $\boldsymbol{X}$, that is connected to the lower level continuous

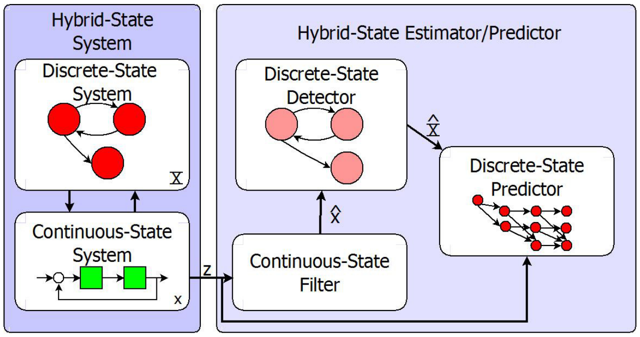


Fig. 1. Module connections for driver-state estimation and prediction.

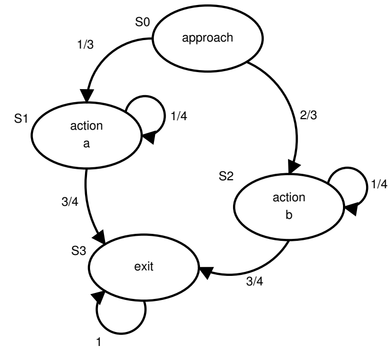


Fig. 2. Simplified FSM model for driver with two possible choices. "Approach" state refers to vehicle approaching an intersection, and exemplary transition probabilities are denoted.

dynamic model of the vehicle, with continuous state, $\boldsymbol{x}$. The continuous state can be any continuous-state system for the generic HSS, but for our applications modeled via a point mass model a la Dubins [7], a bicycle model [8] or a full-car model. The connection between levels are through signals $\boldsymbol{s}$ and $\boldsymbol{S}$, and for the cases that we are interested in, the overall HSS is observable through measurements on the system, $z$, which are functions of the continuous states only. This assumption of the measurement only being functions of the continuous states is a requirement for human-driven vehicles, as an outside observer can detect the vehicle speed or direction, but cannot see the actual decision-making process of the driver.

Earlier studies on hybrid-state system estimation generally involved either a top-down approach, which can be summarized as "estimate the discrete state, use the corresponding continuous model to estimate the continuous state" [9] or a holistic methodology [10]. For our research interests, which involves a vehicle in traffic, direct output on the driver state is scarcely available. For this reason, our method starts from the easier-to-observe states of the lower level, such as vehicle velocity, position and orientation, and builds the estimate of the higher-level state using a real-time estimate of the continuous state. The system connections of this method can be seen in Figure 1.

The information flow in the system can be summarized as follows: Sensing or vehicle-to-vehicle (v2v) communication equipment on the observing, or host, vehicle generates measurements, $z$, on the continuous state of the observed, or target, vehicle. These measurements are filtered through an extended Kalman filter (or similar estimator), the Vehicle Tracking System (VTS)[11], to generate a real-time estimate of the continuous states $\hat{x}$, which is in turn fed into the driver state tracker.

In essence, the HSS is used to represent the interaction between Driver State Dynamics (captured in the DSS), and the Vehicle Dynamics (captured in the CSS). With the HSS structure, two questions arise: 1) How do vehicle dynamics relate to discrete states of the DSS, and 2) How do discrete states of the DSS relate with other DSS states? The former can be looked at as the task of estimation, and the latter, that of prediction.

## III. Driver State Estimation

Two methods for driver state estimation are discussed in this section. The first, rule based estimation, provides a relationship between vehicle dynamics and driver state through a set of rules. With this method, possible driver states in the DSS are defined *a priori*, and changes in vehicle dynamics (such as velocity, acceleration, yaw rate, etc.) govern changes in the driver state under a predefined set of conditions, or rules. The second method, HMM based estimation, provides a stochastic relationship between vehicle dynamics and driver state. With this method, DSS driver states are not predefined, and changes in CSS vehicle dynamics lead to changes in the driver state as determined by training data.

### A. Rule Based Estimation

In rule based estimation, the tracking and estimation of the instantaneous driver state can be accomplished through an internal model of the DSS, which is run by the estimated continuous parameters. A generic and simple driver model FSM developed for demonstration purposes can be seen in Figure 2. The driver chooses between two possible actions in this example FSM, and the vehicle follows the intention of the driver.

With access to the FSM model that governs the decision-making process, the continuous-state estimates, $\hat{x}$, are used to drive a number of FSM state-transition events, either directly, or through secondary calculations on continuous parameters such as distance between vehicles and binary parameters embedded directly into the measurements, such as v2v messages (turn signals or "brake applied" flags), if available. The events are set on continuous parameter estimates, $\hat{x}$, which generate an estimate of the signal, $s$, in order to calculate an instantaneous estimate of the discrete state, $\hat{X}$, through the rules of the FSM model. Further details on the overall system architecture with preliminary results, can be found in [1].

For the FSM model with a higher number of states, as shown in Figure 3, the probability assignment for prediction can be done empirically or through equalizing end-to-end probabilities of possible scenarios. In other words, each state transition that starts at an initial state and ends at a final state is given equal weight, and these end-to-end transitions

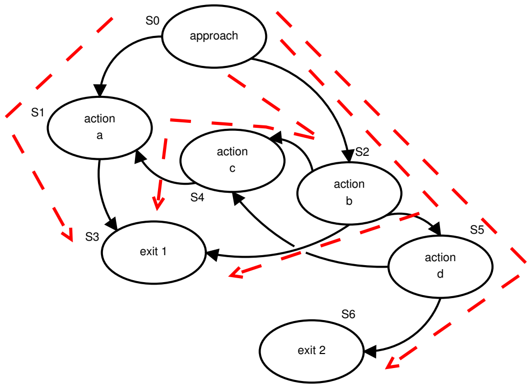

Fig. 3. Seven-state FSM model for driver decisions and end-to-end state-transition paths.

vote on the transitions they occupy. From our seven-state FSM, the transitions from ***S0*** to ***S3*** and ***S0*** to ***S6*** vote or weigh on the transitions they pass through and the number of end-to-end transitions passing through a particular transition determines the likelihood of that edge on the trellis.

As driver models evolve, direct probability assignment becomes time consuming and increasingly prone to human error. In addition, it is not possible to predetermine a set of DSS states that a vehicle must adhere to, nor is it possible to enumerate a set of rules governing the HSS. Formal methodology to generate useful states, and state transition probabilities is a fundamental requirement for the success of the overall estimator implementation, and is discussed in the following section.

### B. HMM Based Estimation

The probabilistic assignments mentioned in the previous section immediately suggest the use of Hidden Markov Models [3], [2] to capture the discrete state dynamics, and, for the development of algorithmic methods to estimate the transition probabilities of the DSS from observations of driver behavior. A discrete HMM consists of a set of $N$ finite "hidden" states, $S_i$, $1 \leq i \leq N$, and a set of $M$ observable symbols per state, $v_k$, $1 \leq k \leq M$. In the development of HMMs for driver behavior estimation, $M$ corresponds to observable CSS parameters and, $N$, the "hidden" DSS states. Additional elements of an HMM are defined as follows (with $q_t$ and $o_t$ denoting the state, and observation symbol, respectively, at time $t$) :

1) The state transition probabilities, $\mathbf{A} = \{a_{ij}\}$, where

$$a_{ij} = P[q_{t+1} = j | q_t = i],\ 1 \leq i,j \leq N$$

In this context, $\mathbf{A}$ corresponds to the transitions probabilities of the DSS in Figure 2 .

2) The observation symbol probability distribution $\mathbf{B} = \{b_j\}$ where,

$$b_j(k) = P[o_t = v_k | q_t = j],\ 1 \leq k \leq M$$

In this context, this distribution defines the relation between the hidden DSS states and CSS observations.

3) The initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = P[q_1 = i],\ 1 \leq i \leq N$$

An HMM can be completely specified by $N$, $M$, and the three probability measures $\lambda = (A, B, \pi)$.

With the aim of uncovering the relationship between continuous observations of driver behavior (such as velocity, orientation, etc.) and the discrete states in the DSS (such as vehicle is performing action a, or action b), HMMs must be trained with actual driving data. Additionally, given an observation sequence, one should be able to evaluate the probability that an observation sequence is explained by a certain model. The latter problem can be solved using a procedure known as the forward procedure to solve $P(O|\lambda)$. Consider the forward variable $\alpha_t(i)$ defined as:

$$\alpha_t(i) = P(o_1 o_2 ... o_t, q_t = i | \lambda) \quad (1)$$

The probability of observing a sequence $O = \{o_1, o_2, ..., o_t\}$ given the model $\lambda$ is:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \quad (2)$$

One can also similarly define a backward variable $\beta_t(i)$ to solve the above problem:

$$\beta_t(i) = P(o_{t+1} o_{t+2} ... o_T, q_t = i | \lambda)$$

One method to train HMMs is to use an estimate of the continuous state variable, $\hat{x}$, and use this estimate to deduce the discrete state in the DSS, $\hat{X}$ using an iterative procedure known as the Baum-Welch method [12]. The Baum-Welch (also called Expectation Maximization) method estimates the maximum likelihood model parameters $\lambda = (A, B, \pi)$ using the definitions of $\alpha_t(i)$ and $\beta_t(i)$ to find updated values of $\bar{\pi}_i$, $\bar{a_{ij}}$, and $\bar{b}_i(k)$.

Once models $\lambda_1, \lambda_2, ..., \lambda_n$, corresponding to $n$ different vehicle actions (such as vehicle turning left, vehicle going straight, etc) have been trained, we can perform state estimation. For a given observation sequence $\mathbf{O} = \{o_1, o_2, ..., o_t\}$, the probabilities $P(O|\lambda_i)$, $i = 1, 2, ..., n$ are calculated using the forward algorithm. The highest likelihood probability corresponds to the estimated vehicle action. Thus, to determine the state at some time, $t$:

$$State(t) = \arg\max_i P(o_1 o_2 ... o_t | \lambda_i)\ i = 1, ...., n \quad (3)$$

An overview of this technique is shown in Figure 4. The final compare step corresponds to equation 3, and $n$ refers to the number of trained HMMs.

## IV. State Estimation Results

Using both of the techniques for driver state estimation, as discussed in the previous section, results obtained validate not only the HSS architecture, but also the rule based estimation and HMM based estimation proposed. While both of these techniques have their relative merits, the HMM based estimation method is more versatile, and requires less external external input.

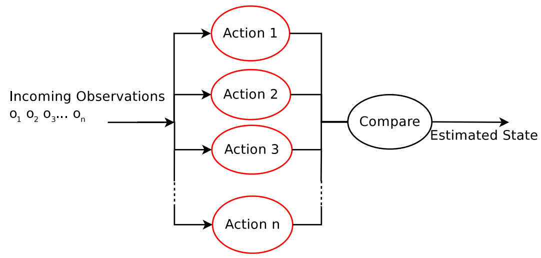


Fig. 4. Choosing the state that 'best' describes **O**

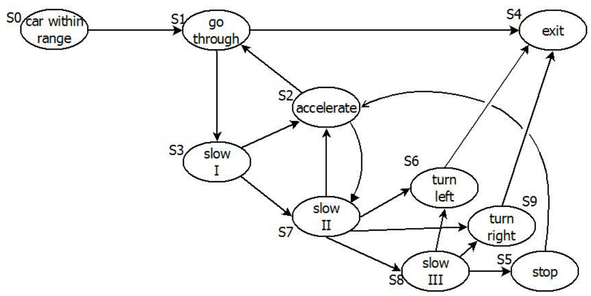


Fig. 5. Rule based FSM

## A. Rule Based Estimation

For the purpose of testing, the FSM used to describe the relation between DSS states is shown in Figure 5. Simple rules for state transitions were derived from vehicle dynamics, or CSS observations. One such example rule is: Transition from ***S1*** to ***S3*** if the speed is less than 25 miles/hour, another is: Transition from ***S3*** to ***S7*** if the speed is less than 15 miles/hour. Possible states were decided based on collected data, and are for describing a vehicle approaching an intersection.

Figure 6 describes a sample estimation of driver state for a vehicle driving in Detroit, MI. Transition probabilities between states were determined by looking at a large set of data collected for vehicles approaching intersections. The topmost plot shows the vehicle distance to intersection, and the second plot shows the vehicle velocity. The third plot shows the current state of the vehicle as determined by a set of pre-determined rules.

## B. HMM Based Estimation

In the process of estimating the discrete state, one HMM is developed for every possible vehicle maneuver. As an example of HMM development for vehicle state estimation, HMMs were developed from experimental observation data, obtained by driving a sensor fitted vehicle in Detroit, MI through a route that contains approximately 75 instances of a vehicle approaching an intersection and either turning left, turning right, continuing straight, and/or stopping before or after one of these actions. The vehicle was fitted with sensors that were capable of reading from the vehicle CAN-bus providing CSS parameters such as velocity, and orientation. An additional GPS sensor provided the location of the vehicle during the test scenarios. From these sensors, continuous estimates, $\hat{x}$, were obtained and vector quantized using K-means clustering to one of $M = 16$ meta-states, $\{o_1, ..., o_{16}\}$,

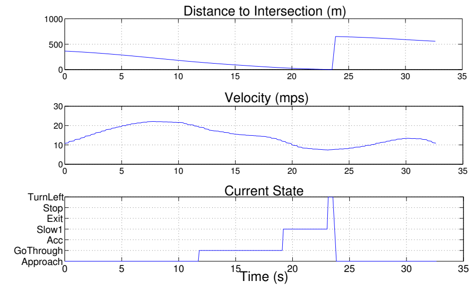


Fig. 6. Estimate and Driver State through Rule Based Approach

which were considered the observation symbols, **O**. Using a modified version of [13], with $N = 5$ possible "hidden" states, $\{S_1, S_2, S_3, S_4, S_5\}$, four individual HMMs were developed for a vehicle approaching an intersection relating to four scenarios are of interest:

1) Action 1 : The vehicle approaches and turns left
2) Action 2 : The vehicle approaches and turns right
3) Action 3 : The vehicle continues straight
4) Action 4 : The vehicle approaches, stops, and proceeds either left, right, or straight

For each of the above scenarios, a corresponding HMM was developed, namely: Left HMM ($\lambda_{LEFT}$), Right HMM ($\lambda_{RIGHT}$), Straight HMM ($\lambda_{STRAIGHT}$), Stop HMM ($\lambda_{STOP}$).

As described previously, in order to estimate the discrete state of a sequence of observations, the observation sequence is compared with each of the four models. The forward algorithm is used to determine which of the four HMMs best describe the observation sequence. The specific technique is slightly modified from figure 4 by using K-means clustering to cluster observations into 1 of 16 clusters.

Results were obtained for multiple runs and compared to ground truth, which was was obtained by plotting the vehicle trajectory in Google Maps. Of the four HMMs, the HMM that resulted in the highest probability of the supplied state sequence was said to describe the vehicle maneuver "best". Mathematically, state estimation was said to be correct when the model that maximized posterior probability $P(o|\lambda)$ for the full length sequence matched with ground truth. The results obtained for 20 sample observation sequences is shown in Figure 7.

Figure 7 gives the estimate values for five individual vehicle observations of four vehicle maneuvers at an intersection. The data presented describes the estimators output at a given time, $t$, as described in Equation 3. Thus, for the top figure relating to a "Left Turn Sequence," five test observation sequences with ground truth corresponding to a vehicle turning left was run through the four developed HMMs. At every time step, equation 3 was evaluated, and the HMM that best described the sequence until that point was the estimated state. This procedure was similarly repeated for five observation sequences with ground truth corresponding to a vehicle stopping and turning left, a vehicle turning right, and a vehicle going straight through an intersection. The results of Figure 7 show correct recognition for these 20 observation sequences. Additionally, the system cor-

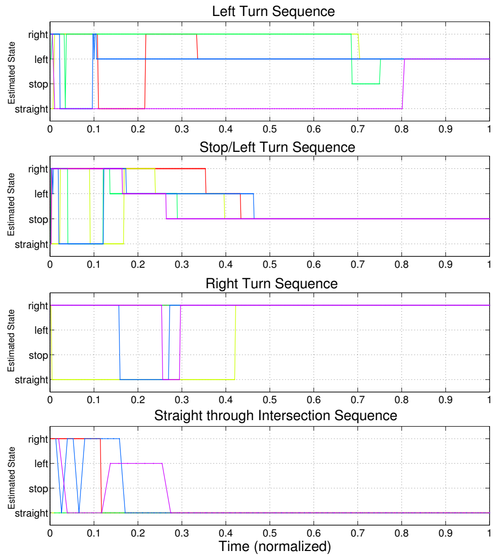


Fig. 7. State Estimation for 5 different vehicle runs relating to: a) Left Turn Sequence, b) Stop and Left Turn Sequence, c) Right Turn Sequence, and d) Straight through intersection Sequence

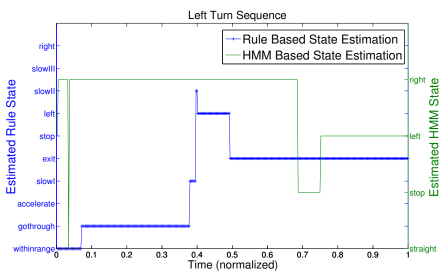


Fig. 8. Rule based estimation vs. HMM based estimation: Vehicle Turning and Stopping

rectly recognizes events described by two different models - namely, Stop and Turns. Even with the limited availability of driving data, the outlined approach provides a mathematical foundation for the relationship between HSS components.

### C. Comparison of the Two Techniques

Both methods were tested with 40 observation sequences. Of the 40 tests, both methods produced the same result (left, right, or straight), 29 times. Of the 11 runs where results did not match, in 9 of these sequences, the rule based approach incorrectly identified the sequences, whereas the HMM based approach correctly identified the sequences. For each of these 9 observation sequences, the Rule Based Approach estimated a right turn, whereas the ground truth is a left turn. From the FSM in Figure 5, the cause for this discrepancy can be determined. Since no transitions between left (*S6*) and right (*S9* states is allowed, if a vehicle is moving to the right before taking a left turn, it will still fall into the right turn state. Additionally, consider the case where a vehicle is turning and stopping. The rule based approach will only show a left turn, since no transition path from *S6* to *S5* is available, or transitions from *S5* to *S6* occur because conditions for a state change are not met. Such an example is presented in Figure 8

The left axis refers to the rule based states, and the right axis refers to state estimated from the HMM based

| Sequence | Normalized $T_{RULE}$ | Normalized $T_{HMM}$ |
|---|---|---|
| 1 | 0.2966 | 0.4100 |
| 2 | 0.1433 | 0.1418 |
| 3 | 0.1966 | 0.2253 |
| 4 | 0.4306 | 0.5604 |
| 5 | 0.3587 | 0.1090 |

TABLE I

NORMALIZED TIMES FOR CORRECT IDENTIFICATION

approach. Another interesting comparison, is in the amount of observation time taken by each approach to correctly identify the observation sequence. Table I gives five left turn observation sequences, along with the normalized time taken by each of the techniques in correctly identifying the sequence. The time taken by both approaches depends largely on when certain observations are made (for example, a large yaw rate early in the sequence will lead to a fast turn detection). While one may argue that the drawback of the rule-based method is merely a flaw in the transition definitions, given the large number of possible states, it will be very difficult to have a all-encompassing rule based system that enumerates, accurately, every possible transition, further motivating the HMM approach.

## V. FUTURE STATE PREDICTION

In order to predict future driver states(s), it is necessary to accurately estimate the current state based on information about the driver state and vehicle dynamics. In addition, it is necessary to estimate state transition probabilities since the predictor module of the overall architecture utilizes a FSM model and state-transition probabilities. These probabilities can be deduced either through empirical reasoning or statistical reasoning of real life data. While the former approach provides an acceptable starting point, it is difficult when real-life data is hard to collect and analyze. To overcome these limitations, we propose using a statistical tool called Hierarchical Hidden Markov Models to determine the state transition probabilities.

### A. Hierarchical Hidden Markov Models

HHMMs [5] are multi-level HMMs and generalize the concept of HMMs by making each 'hidden' state an HMM. HHMMs are special cases of the more general dynamic Bayesian networks [14], and have a strong connection to Switching Linear Dynamical Systems. In the application of predicting driver behavior, a state such as 'action n' (Figure 4) is itself an HMM. By developing HMMs for a variety of vehicle states (or actions), HHMMs can be used to describe the interaction between each HMM. In a simple example where a driver has two possible choices such as in Figure 2, "action a" and "action b" are HMMs, and the relationship between them will be determined by the HHMMs.

Figure 9 shows the general structure of a HHMM. Using the HSS architecture, the observations, $y_i$ correspond to CSS observations. The latent, or hidden, variable, $x_i$, corresponds to HMM states as developed in the previous section. The highest state, $S_i$, corresponds to a higher level HMM that

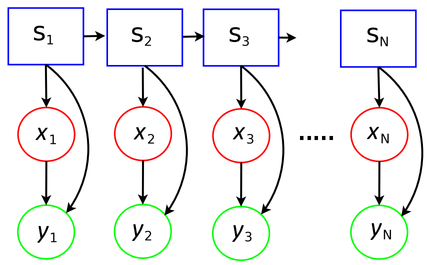


Fig. 9. Structure of HHMM

describes the relationship between the lower HMM states, $x$. With HHMM formalization, the relationship between discrete states in the DSS can be determined.

### B. Prediction

Once transition probabilities between discrete states have been determined, one can predict future states with the knowledge of present state and state transition probabilities. The prediction stage of the proposed system utilizes a representation structure called a *Trellis*, which has been used extensively in communication applications through the Viterbi Algorithm [15]. The trellis is a representation of a finite state machine, just like the state diagram, that makes probabilistic calculations and predictions easier. The correspondence between the state diagram and the trellis is demonstrated in Figures 2 and 10 is shown as:

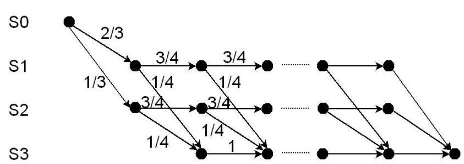


Fig. 10. Trellis representation for the simple FSM model.

Previous work on prediction has made use of transition probabilities derived empirically from real-life driving. A successful example of prediction can be seen in Figure 11.

## VI. Conclusions and Future Work

We presented a novel architecture for describing the coupling of vehicle and driver through the Hybrid-State system. While previous work [1] has relied on a rule based method, the main contribution of this paper is a new mathematical formulation of the interaction between the HSS hierarchy. In order estimate the state of a vehicle, we used HMMs, and propose extending to HHMMs to predict the future states of a vehicle. At first glance, some of the work presented in this may seem similar to [4], but a fundamental difference between the approach proposed here is the vehicle/driver coupling and connection with system architecture of the HSS. The extension to HHMMs is a novel extension that will allow for automated estimation and prediction of vehicle state.

In order to improve the results as described in Figure 7, rigorous data collection is required. Proposed data collection involves recruiting participants to drive a sensor fitted vehicle in one hour predefined routes on arterial roads in Columbus, OH. The CPS group has access to vehicles fitted with high resolution and high frequency sensors that will improve the results and allow the development of HMMs for a large set of vehicle maneuvers (n). With HMMs developed for a large set of maneuvers, they will be extended to HHMMs. With the development of HHMMs for a single vehicle, it will be possible to predict vehicle behavior after formulation of a trellis as in Figure 10. Future extensions also include modeling swarms of vehicles, which may be autonomous and/or human controlled, dynamically forming "teams" or convoys.

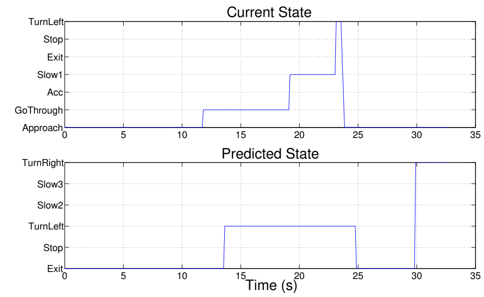


Fig. 11. Successful prediction of Left Turn

## References

[1] A. Kurt, J. Yester, Y. Mochizuki, and U. Ozguner, "Hybrid-state driver/vehicle modelling, estimation and prediction," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010, pp. 806–811.

[2] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSp Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[3] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[4] D. Mitrovic, "Driving event recognition by hidden Markov models," in *Telecommunications in Modern Satellite, Cable and Broadcasting Services, 1999. 4th International Conference on*, vol. 1. IEEE, 2002, pp. 110–113.

[5] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Machine learning*, vol. 32, no. 1, pp. 41–62, 1998.

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

[7] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[8] R. S. Hand, "Comparisons and stability analysis of linearized equations of motion for a basic bicycle model," Master's thesis, Cornell University, Ithaca, NY, 1988.

[9] A. Balluchi, L. Benvenuti, M. Di Benedetto, and A. Sangiovanni-Vincentelli, "Design of observers for hybrid systems," *Hybrid Systems: Computation and Control*, pp. 59–80, 2002.

[10] G. Ferrari-Trecate, D. Mignone, and M. Morari, "Moving horizon estimation for hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 47, no. 10, pp. 1663–1676, 2002.

[11] K. Streib, U. Ozguner, J. Martin, Y. Mochizuki, and K. Ishikawa, "A sensor based assessment of imminent collisions at right angle intersections," in *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, 2008, pp. 210 –215.

[12] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[13] K. Murphy, "Hidden markov model (hmm) toolbox for matlab," *online at http://www. ai. mit. edu/˜ murphyk/Software/HMM/hmm. html*.

[14] D. Barber and A. Cemgil, "Graphical Models for Time-Series," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 18–28, 2010.

[15] J. Forney, G.D., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268 – 278, 1973.