Highway Traffic Modeling Using Probabilistic Petri Net Models

Keyu Ruan, Lingxi Li, and Yaobin Chen

***Abstract*— In this paper, we propose a novel method for modeling the highway traffic using Probabilistic Petri nets (PPNs). More specifically, the highway has been partitioned into discrete segments and probabilistic measures are derived based on the traffic data considering the vehicle movements. The proposed model is validated through the study of a publicly available dataset called Active Transportation Demand Management (ATDM) Trajectory Level Validation, which provides the real traffic data from different driving scenarios. The proposed method will generate graphical structures of the PPN as well as all important attributes related to the real traffic data. The output model can be used for path planning and collision avoidance for highway traffic.**

***Index Terms*— Probabilistic Petri nets, highway traffic, reachable markings.**

I. INTRODUCTION

With the fast development of intelligent vehicle technologies, vehicles are becoming smarter with advanced sensing and control units. Various types of techniques have been integrated into these intelligent vehicles. In [1], a technique was introduced for guidance control of the parallel parking, where unskilled drivers can benefit from it in parking tasks. In [2], the authors studied the strategies of decision-making and driving state control of unmanned vehicles. The objectives were achieved with interval type-2 fuzzy sets, fuzzy comprehensive evaluation, and fuzzy control rules. Authors in [3] surveyed the topics on traffic signal management and control using Model Predictive Control (MPC) algorithms. Driver behavior data were collected and analyzed in [4] for the evaluation of an in-vehicle camera-based driver state sensing system that can be an objective and reliable source for improving driving safety.

Other than these systems, intelligent vehicles can also provide braking assistance to ensure occupant safety. For example, the Automatic Emergency Braking (AEB) systems, such as Crash Imminent Braking (CIB) system and Dynamic brake support (DBS) system, are believed to be technologies representing the significant advances in vehicle safety. The evaluation of the CIB system [5] showed that the system can apply automatic braking to mitigate damages and avoid potential collisions. It is proved that these systems operate very well within a certain speed range towards pedestrians and bicyclists, in particular, the braking behavior of pedestrian AEB systems was analyzed in [6]. The authors of [7] evaluated Road Departure Prevention systems (RDPSs), which can potentially reduce the crash risk by issuing visual/audible warning signals to the driver and engage automatic steering adjustment when the vehicle is detected to deviate from the current lane or the road edge. This system works well with clear lane/road edge markings and/or roadside materials such as grass.

Furthermore, a variety of algorithms and technologies that can provide vehicles with more advanced capabilities were developed. The authors in [8] have proposed an approach for estimating the rotated angle of the steering wheel. A particle filtering algorithm was used for the estimation. By applying deep learning-based regression framework, the patterns of normal driving and driving toward an obstacle have been evaluated. The work in [9] focused on obtaining a control structure that is used for stabilizing the vehicle when facing emergency scenarios. The structure prioritizes collision avoidance among conflicting objectives, which is implemented using model predictive and feedback controllers.

With the developed advanced sensing and driver-assistance systems, it is feasible for intelligent vehicles to make decisions to reduce the risk of crashes based on the traffic and road conditions. Many researchers have been investigating this issue from the viewpoint of traffic engineering. For instance, the authors in [10] have proposed a method for highway segmentation with a speed profile. With some appropriate merging algorithms, these highway segments could be used for trip planning, infrastructure management, and other decision-making tasks. In [11], the authors discretized the highway into overlapping areas with equal distance. Besides, they also discretized the speed into fixed intervals. Another approach has been proposed in [12], which is used for optimization of the sensor placement on highway for a better observation of traffic conditions.

The idea of road discretization is not new. However, different approaches have been used for different applications. There is still much room to model and simulate the traffic system in a more complete way. In this paper, we propose a probabilistic Petri net model for studying the driving behavior of vehicles on each lane of a certain highway segment. This method is quite general and can be adopted to different highways with different geometries and traffic volume. In particular, the contributions of this paper are summarized as follows.

- The highway segment is discretized into blocks that are homogeneously distributed in terms of distance.
- Driving behavior data of vehicles are classified in terms of time. The original driving dataset is studied with a

The work in this paper was supported in part by the QAII Joint Fund Project 16-7-1-4-JCH.

Keyu Ruan, Lingxi Li, and Yaobin Chen are with the Transportation and Autonomous Systems Institute (TASI), Department of Electrical and Computer Engineering, Indiana University-Purdue University-Indianapolis, 723 West Michigan Street, SL-160, Indianapolis, IN, USA. Emails: {keyuruan, LL7, ychen}@iupui.edu.

reasonable sampling rate.

- The Probabilistic Petri net model is developed by fitting the driving behavior data into the discretized highway blocks. The movement of vehicles will follow a certain probability assigned by the model according to the traffic data available.

II. PRELIMINARIES AND NOTATIONS

In this section, we will introduce the basic terminologies that are used throughout this paper. More details about Petri nets can be found in [13], [14].

A Petri net $N_P = (P, T, Arcs, Weights)$ is a graph with directed arcs connecting from places to transitions (and transitions to places), where $P = \{P_1, P_2, \ldots, P_n\}$ represents the set of *places* (shown as empty circles) with a finite size n, $T = \{t_1, t_2, \ldots, t_m\}$ represents the set of *transitions* (shown as empty or solid bars) with a finite size m, $Arcs$ is a set of arcs that connect P and T, and $Weights : Arcs \to \{1, 2, 3, \ldots\}$ represents the *weighting function* of the arcs.

Notation I is used to represent the incident matrix of the Petri net and I is defined as $I = I^{out} - I^{in}$, where I^{out} is called the output incident matrix of the Petri net and I^{in} is called the input incident matrix of the Petri net. Each entry I_{ij}^{out}, which is at the i^{th} row, j^{th} column position of matrix I^{out}, denotes the weight of the arc from transition t_j to place P_i; Similarly, each element I_{ij}^{in} in matrix I^{in} denotes the weight of the arc that starts at place P_i and ends up at transition t_j, where $i \in [1,\ n]$ and $j \in [1,\ m]$. Note that all weights on the arcs should be non-negative integer numbers and the value of I_{ij}^{out} (or I_{ij}^{in}) is set to be zero if there is no arc connecting place P_i and transition t_j. Hence, we can also use $N_p = (P, T, I^{out}, I^{in})$ to represent a Petri net.

$M : Places \to Z_0^+$ is a vector used to denote the marking, which assigns a non-negative integer number of tokens into each place of the Petri net. The tokens are represented as black dots in a Petri net graph. The total number of tokens in all places is denoted as $|M|$ (i.e., $|M| = \sum_{i=1}^{n} M(P_i)$). A transition t is *enabled* when each of its input place P_{in} has no less than $I^{in}(P_{in}, t)$ tokens, where $I^{in}(P_{in}, t)$ represents the arc weight from place P_{in} to transition t. $M[t\rangle$ is used to represent that transition t is enabled at marking M. The enabled transition t can fire. When transition t fires, $I^{in}(p, t)$ tokens should be removed from all of its input places, while all of its output places should be deposited with $I^{out}(p, t)$ tokens. This process will yield a new marking, denoted by M' that satisfies $M' = M + I(:, t)$, where $I(:, t)$ denotes the column in the incident matrix I that corresponds to transition t. This is also denoted by $M[t\rangle M'$, and we say marking M' is *reachable* in one single firing step from marking M via the firing of transition t. We use a vector $v : transition \to Z_0^+$ to denote the firing vector that has a non-negative integer number in each of the entry, which represents the number of firings for each transition t. The state equation $M'' = M + I * v$ is an extension of $M' = M + I(:, t)$, where M'' is also a *reachable* marking from M in multiple firing steps.

Based on the study in [15], the probabilistic Petri nets have the similar structures with regular Petri nets except the firing of each transition is associated with a probability, which captures its likelihood to occur. In particular, a probabilistic Petri net is a five-tuple $N_pp = (P, T, Arcs, Weights, prob)$, where $P, T, Arcs$, and $Weights$ are the same as regular Petri nets, the probability function $prob$ is defined as $prob : T \to [0,\ 1]$, which assigns each transition a number between 0 and 1 for its likelihood to occur if it is enabled.

Fig. 1 shows an example of a probabilistic Petri net structure. Note that if all transitions share the same input place, the sum of probabilities of all these transitions is one. In Fig. 1, for instance, $prob_1 + prob_2 + prob_3 = 1$.

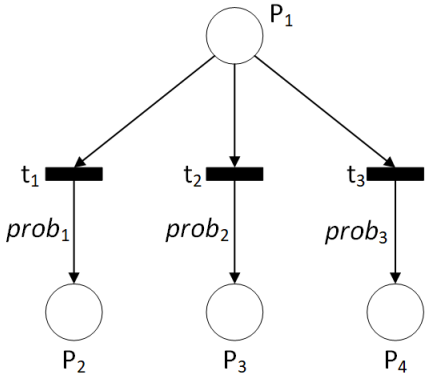

Fig. 1. An example of a probabilistic Petri net.

III. TRAFFIC DATA PROCESSING AND SEGMENTATION

The dataset we study is called ATDM Trajectory Level Validation [16] by the U.S. Department of Transportation, which provides the information of vehicles driving on local roads and highway. It includes a variety of data attributes such as *datetime*, *latitude*, *longitude*, *speed*, *acceleration*, and *laneID*.

A. Data Filtering

The data attributes needed for our analysis are *latitude*, *longitude*, *speed*, *laneID*, and *testing datetime*. Note that the recorded data include data points for both local and highway scenarios. Since we are only interested in the driving scenarios on highway, the data points with inappropriate speed will be removed.

Furthermore, the fast data sampling rate of the original dataset is not suitable for our case because the short time period will not generate much difference in vehicle position and speed. Hence, we need to find an appropriate sampling ratio. From the work in [11], the suggested time interval is 1 to 3 seconds. In our work, we select the lower bound of the suggested values, which is 1 second.

B. Highway Segment Discretization

Since the original database does not provide the detail of the driving route, road geometry, and traffic conditions, there are two key assumptions in this study: A1) only straight highway geometry is considered, A2) there is no heavy traffic or congestion that will significantly affect the driver's

intention. The reason we make these assumptions is for our calculations. Specifically, we choose to calculate and accumulate the traveling distance between data points with latitudes and longitudes to estimate the length of the highway segment. Because the time gap between two consecutive data points is very short, the accumulated distance could represent the length of highway segment, regardless of the highway geometry. Under these two assumptions, it is feasible to determine the total length of the highway. Calculating the distance that each vehicle traveled between two data points is straightforward. Equations (1)–(3) below show the Haversine formulas [17], which is used to calculate the distance d with latitude and longitude:

$$a = sin^2(\frac{\Delta\phi}{2}) + cos\phi_1 \cdot cos\phi_2 \cdot sin^2(\frac{\Delta\lambda}{2}), \tag{1}$$

$$c = 2 \cdot atan2(\sqrt{a}, \sqrt{1-a}), \tag{2}$$

$$d = R \cdot c, \tag{3}$$

where ϕ_1, ϕ_2, $\Delta\phi$, $\Delta\lambda$, and R represent the two latitudes, the difference of the latitudes (in radians), the difference of the longitude (in radians), and the radius of the earth, respectively. Furthermore, we can find the longest data segment from the filtered dataset. The total length of the highway could be considered as the accumulated distance of the longest data segment.

The average length of common compact cars in the U.S. is from 4.3 to 4.6 meters, while the larger-sized cars could be 0.2 to 0.5 meters longer. Hence, it is reasonable to make the length of each discretized segment to be 10 meters, which is about the length of two vehicles. The distance is practical since it is close enough for the crash imminent braking (CIB) system to be triggered (CIB triggering distance is from 5 to 15 meters as suggested in [5]), while still leaves room for the following vehicles to avoid potential collision. With the information of the total length of the highway of interest and the length of proposed discretized segment, it is not difficult to create the discretized model.

Definition 1 A real number is assigned to each transition that represents its *Reward*, denoted by r.

The *Reward* is used to represent the level of occupation of the following place. Different from *prob*, *Reward* is a quantity that is correlated with the emptiness of the following place, which can be used to indicate the potential traffic safety and more intuitive than *prob*. The value of *Reward* is calculated from the probability *prob* that one vehicle moving from one block to another, as shown in Equation (4) below:

$$r = -log_{10}(prob). \tag{4}$$

The method to obtain the *prob* values for all transitions in the Petri net is through the prepared dataset. It is not difficult to know which place that each data point belongs to since we have calculated the accumulated distance. Therefore, we have the distributions of the whole dataset for all places. Then, the *prob* values are calculated with the *Monte Carlo* method. Starting from one certain place, excluding the ones with zero probability, we can have the set of places that the token would be transited to. By doing some preliminary statistical analysis, the probabilities of transiting to these output places can be easily obtained.

Example 1 Consider a highway segment with three lanes shown in Fig. 2. This highway segment can be discretized to generate the Probabilistic Petri net. As shown in the right figure in Fig. 2, each partitioned region is treated as a place in the Petri net. Vehicles driving on the highway is considered to as the token movement. Transitions t_1, t_2, and t_3 are used to connect places. We assume that there are two data points in place P_{11} and one data point in place P_{13}. Therefore, the probability values satisfy $prob_1 = prob_2 = 0.5$ since two tokens in P_{11} are transited to P_{22} and P_{32}, respectively. On the other hand, the probability $prob_3$ is equal to 1 since the only token in P_{13} is transited to P_{31}. Calculating the r values using Equation (4), we can obtain the resulting probabilistic Petri net shown in Fig. 3.

In Fig. 3, the places that have not involved in the transition firings are not shown in order to simplify the structure. We can see that P_{22} places and P_{32} obtained one token each from the two tokens in place P_{11}. Hence, the probabilities of t_1 and t_2 are both 0.5. Similarly, since the only token in P_{13} has been transited to P_{31}, the probability of t_3 is 1. The reward values are also listed as $r_1 = r_2 = 0.301$ and $r_3 = 0$. Since $prob \in [0,\ 1]$, $-log_{10}(prob)$ is always non-negative and is monotone decreasing as the value *prob* increases. Hence, it is equivalent to find the smallest *prob* value or to obtain the largest r value. The corresponding incident matrix I of the Petri net in Fig. 3 is shown in Equation (5) below:

$$I = \begin{array}{c} P_{11} \\ \cdots \\ P_{13} \\ \cdots \\ P_{22} \\ \cdots \\ P_{31} \\ P_{32} \\ \cdots \end{array} \begin{bmatrix} -1 & -1 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & -1 \\ \cdots & \cdots & \cdots \\ 1 & 0 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ \cdots & \cdots & \cdots \end{bmatrix}. \tag{5}$$

Note that the constructed Petri net in Fig. 3 has transitions with a single input and a single output, which indicates that each column of incident matrix I has at most two non-zero entries with values ± 1. The columns might also contain only zeros when there exists self-loops.

IV. Driving data fitting

After obtaining the Petri net with reward values, it is feasible to fit the driving data in the database to the model. Note that we consider only the highway scenarios, there are one constraint for the token moving: The tokens on all lanes have to move synchronously. Since the Petri net model is used to simulate the highway traffic and the vehicles cannot

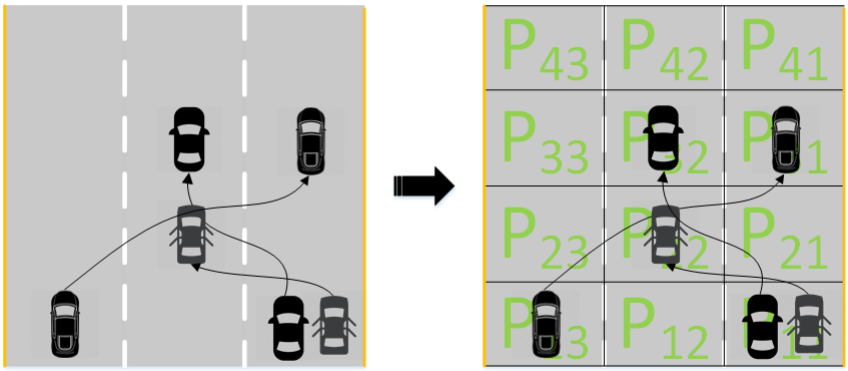

Fig. 2. An example for highway segment discretization.

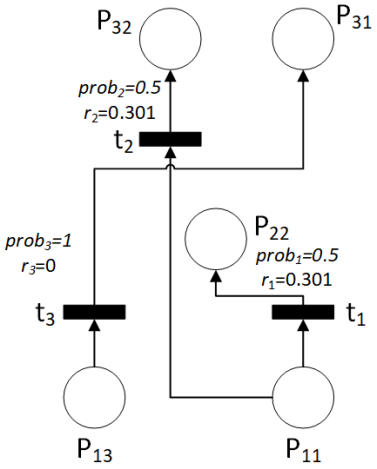

Fig. 3. The probabilistic Petri net obtained from Fig. 2.

stop on the highway, the tokens representing vehicles that are regulated to move synchronously. In order to achieve the constraint in Petri net, the tokens will be transited in sequence, which means only when all tokens are transited once, the newly generated marking is considered as a reachable marking.

We use Fig. 2 to illustrate this idea. Initially, we can see that two lanes have vehicles. Since the vehicles should move synchronously, in one phase, we need to move the vehicle in place P_{11} to place P_{22} or P_{32} and move P_{13} to place P_{31}, according to the possible transition relationship.

Known the idea above, now we represent the constraint into the state equation of Petri net in Equation (6) below:

$$M_j = M_i + I \cdot v, \;\; i, j = 0, 1, 2, \ldots \tag{6}$$

Expanding Equation (6) into the matrix form, we have Equation (7) below:

$$\begin{bmatrix} M_{j1} \\ M_{j2} \\ \ldots \\ M_{jn} \end{bmatrix} = \begin{bmatrix} M_{i1} \\ M_{i2} \\ \ldots \\ M_{in} \end{bmatrix} + \begin{bmatrix} I_{11} & I_{12} & \ldots & I_{1m} \\ I_{21} & I_{22} & \ldots & I_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ I_{n1} & I_{n2} & \ldots & I_{nm} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \ldots \\ v_m \end{bmatrix} \tag{7}$$

Applying the constraint to the Petri net in Example 1, the firing vector v should always have two non-zero entries to represent the synchronously moving traffic on the two lanes, which means:

$$\sum_{index=1}^{m} v_{index} = 2. \tag{8}$$

The non-zero v_{index} pairs can be determined from M_i, which can be shown in Example 2.

Example 2 For the Petri net in Example 1, consider the initial marking M_0 shown in Equation (9) below:

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ \ldots \\ 0 \end{bmatrix}. \tag{9}$$

By inspecting the constructed incident matrix shown in Equation 5, we can find the negative entries in rows corresponding to places P_{11} and P_{13}. For P_{11}, negative entries are in columns 1 and 2. For P_{13}, negative entry is in column 3. Hence, transitions t_1, t_2,, and t_3 can fire. The corresponding firing vector and reward values are given by:

$$v = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \; r = -log_{10}^{0.5 \cdot 1} = 0.301, \tag{10}$$

or

$$v = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \; r = -log_{10}^{0.5 \cdot 1} = 0.301. \tag{11}$$

Therefore, the calculated set of markings M_1 include:

$$M_1 = \begin{matrix} P_{11} \\ P_{12} \\ \dots \\ P_{22} \\ \dots \\ P_{31} \\ \dots \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \\ \dots \\ 1 \\ \dots \end{bmatrix}, \quad or \quad M_1 = \begin{matrix} P_{11} \\ P_{12} \\ \dots \\ P_{31} \\ P_{32} \\ \dots \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \\ 1 \\ \dots \end{bmatrix}. \quad (12)$$

The calculated markings can be used as the markings for further calculations. By doing the iterative calculations, all reachable markings that represent the traffic environment can be obtained. They are stored in different layers based on the time instants of their firings from the initial marking M_0.

V. Data Fitting Results

In this section, we use the method proposed above to discretize the highway segment and model the driving behavior in the dataset. In order to simplify the calculations, We use data from the first three lanes of the dataset. We apply the data filtering and highway discretizing method introduced in Section III so that we can obtain the features of the filtered database and discretized highway, summarized in Table I below.

TABLE I

Features of the filtered database and discretized highway.

Features of the dataset	Value
#Filtered data segments	1,660
#Data points	24,600
Time gap (s)	1
Features of the highway	**Value**
Total length (m)	1,261.3
Length of each block (m)	10
#Lanes	3
#Blocks	381

Note that each filtered data segment is treated as a separate driving circuit in our study. The number of blocks will be equal to the number of places in the Petri net model that we are developing. Furthermore, by fitting the data points into discretized blocks, we can obtain the values of the reward of each transition. In the end, there are 314 places and 818 transitions in this Petri net model since some places without any connections to others have been removed. Because of the size limitation, Fig. 4 shows only the first three layers of the generated Petri net. The reward values of corresponding transitions are provided in Table II.

Once the Petri net has been generated, we can obtain its reachable markings, as illustrated in Section III and Section IV. The total number of reachable markings is 1,984 with 49 layers. Because of the space limitations, we only list the number of markings in each layer in Table III. Note that each of these markings can be generated from different paths with different probabilities and different reward values. All these feasible paths need to be explored to deal with path planning problems using the Petri net model.

TABLE II

Reward values of transitions in Fig. 4.

Transition	Reward	Transition	Reward
t_{73}	0.1146	t_{455}	5.7944
t_{74}	4.0540	t_{456}	8.6018
t_{75}	6.2416	t_{457}	9.6018
t_{76}	8.4115	t_{516}	0.5595
t_{119}	0.0439	t_{517}	1.6938
t_{120}	7.2288	t_{518}	9.3376
t_{121}	5.4215	t_{519}	9.3376
t_{166}	2.2305	t_{520}	9.3376
t_{167}	0.3831	t_{521}	8.3376
t_{168}	7.2192	t_{702}	4.1293
t_{169}	6.2192	t_{703}	3.5443
t_{206}	7.7649	t_{704}	0.2713
t_{207}	1.0784	t_{705}	5.1293
t_{208}	0.9705	t_{729}	0
t_{209}	6.7649	t_{752}	3.7004
t_{210}	8.7649	t_{753}	0.2410
t_{422}	0.0240	t_{754}	4.7004
t_{423}	6.5580	t_{755}	4.7004
t_{424}	8.1430	t_{773}	4.3923
t_{425}	8.7279	t_{774}	0.5850
t_{454}	0.0319	t_{775}	1.8074

TABLE III

The number of reachable markings in each layer.

Layer	Number	Layer	Number
1	1	26	44
2	16	27	48
3	69	28	44
4	76	29	44
5	70	30	44
6	66	31	32
7	68	32	24
8	72	33	36
9	66	34	28
10	63	35	28
11	66	36	28
12	62	37	24
13	62	38	20
14	63	39	12
15	63	40	16
16	54	41	16
17	55	42	8
18	56	43	12
19	44	44	12
20	60	45	12
21	68	46	12
22	60	47	4
23	60	48	4
24	48	49	4
25	40		

VI. CONCLUSIONS

In this paper, we proposed a method to discretize the highway segment and model driving behaviors with probabilistic Petri nets. This method discretized the highway into many blocks. By fitting real-time traffic data, a probabilistic Petri net structure was developed, with places representing blocks on the highway, transitions representing driving behavior, and tokens representing vehicle movement. Reward values are associated to each transition based on the probability of each transition firing. After Petri net model is developed, the reachable markings can be obtained. The calculation is a bit different from the traditional Petri net concept because we require the tokens move synchronously to model traffic

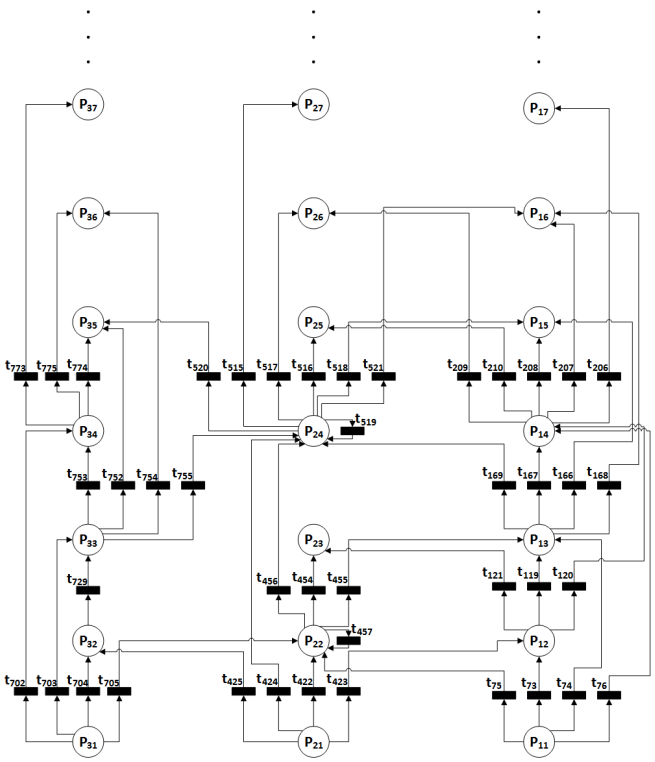

Fig. 4. Part of the Petri net generated from the discretized highway.

situation in a more realistic way. This work can be extended in several ways. The developed model can be used to investigate path planning algorithms for collision avoidance on highway. It is also interesting to explore more traffic datasets to make the model more realistic.

REFERENCES

[1] J. Tan, C. Xu, L. Li, F.-Y. Wang, D. Cao, and L. Li, "Guidance control for parallel parking tasks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 301-306, January 2020.

[2] X. Zhao, K. Yan, H. Mo, and L. Li, "Type-2 fuzzy control for driving state and behavioral decision of unmanned vehicle," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 178-186, January 2020.

[3] B.-L. Ye, W. Wu, K. Ruan, L. Li, T. Chen, H. Gao, and Y. Chen, "A survey of model predictive control methods for traffic signal control," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 623-640, May 2019.

[4] R. Tian, K. Ruan, L. Li, J. Le, J. Greenberg, and S. Barbat, "Standardized evaluation of camera-based driver state monitoring systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 716-732, May 2019.

[5] S. Chien, M. T. Moury, G. Widmann, W. Kosiak, L. Li, and Y. Chen, "Performance measurement of vehicle Crash Imminent Braking systems", in *Proc. 2011 IEEE International Conference on Vehicular Electronics and Safety,* pp. 107-112, Beijing, China, July 2011.

[6] A. Lopez, R. Sherony, S. Chien, L. Li, Q. Yi, and Y. Chen, "Analysis of the braking behaviour in pedestrian automatic emergency braking", in *Proc. 2015 IEEE 18th International Conference on Intelligent Transportation Systems,* pp. 1117-1122, Canary Islands, Spain, September 2015.

[7] D. Shen, Q. Yi, L. Li, S. Chien, Y. Chen, and R. Sherony, "Data Collection and Processing Methods for the Evaluation of Vehicle Road Departure Detection Systems", in *Proc. 2018 IEEE Intelligent Vehicles Symposium (IV),* pp. 1373-1378, Changshu, Suzhou, China, July 2018.

[8] V. John, A. Boyali, H. Tehrani, K. Ishimaru, M. Konishi, Z. Liu, and S. Mita, "Estimation of steering angle and collision avoidance for automated driving using deep mixture of experts," *IEEE Transactions on Intelligent Vehicles,* vol. 3, no. 4, pp. 571–584, December 2018.

[9] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Transactions on Control Systems Technology,* vol. 25, no. 4, pp. 1204–1216, July 2017.

[10] R. Aziz, M. Kedia, S. Dan, S. Sarkar, S. Mitra, and P. Mitra, "Segmenting highway network based on speed profiles," in *Proc. IEEE 18th International Conference on Intelligent Transportation Systems,* pp. 2927–2932, Canary Islands, Spain, September 2015.

[11] Y. Guan, S. E. Li, J. Duan, W. Wang, and B. Cheng. "Markov probabilistic decision making of self-driving cars in highway with random traffic flow: A simulation study." *Journal of Intelligent and Connected Vehicles*, vol. 1, no. 2, pp. 77–84, August 2018.

[12] S. Contreras, P. Kachroo, and S. Agarwal, "Observability and sensor placement problem on highway segments: A traffic dynamics-based approach," *IEEE Transactions on Intelligent Transportation Systems,* vol. 17, no. 3, pp. 848–858, March 2016.

[13] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems,* New York: Springer, 2008.

[14] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE,* vol. 77, no. 4, pp. 541–580, April 1989.

[15] Y. Emzivat, B. Delahaye, D. Lime, and O.H. Roux, "Probabilistic time Petri nets", In: Kordon F., Moldt D. (eds) Application and Theory of Petri Nets and Concurrency. PETRI NETS 2016. Lecture Notes in Computer Science, vol 9698. Springer.

[16] U.S. Department of Transportations (USDOT) Intelligent Transportation Systems (ITS) Joint Program Office (JPO), "Active Transportation Demand Management (ATDM) Trajectory Level Validation", https://data.transportation.gov/Automobiles/Active-Transportation-Demand-Management-ATDM-Traje/25r8-p3cy, November 2018.

[17] K, G, Arthur and K. Theresa M, "Appendix B: B9. Plane and Spherical Trigonometry: Formulas Expressed in Terms of the Haversine Function", *Mathematical handbook for scientists and engineers: Definitions, theorems, and formulas for reference and review (3 ed.),* Mineola, New York, USA: Dover Publications, Inc. pp. 892-893, 2000.