

# Learning to Segment and Represent Motion Primitives from Driving Data for Motion Planning Applications

Boyang Wang<sup>1,2</sup>, *Student Member, IEEE*, Jianwei Gong<sup>1</sup>, *Member, IEEE*, Ruizeng Zhang<sup>1</sup>, Huiyan Chen<sup>1</sup>

**Abstract**—Developing an intelligent vehicle which can perform human-like actions requires the ability to learn basic driving skills from a large amount of naturalistic driving data. The algorithms will become efficient if we could decompose the complex driving tasks into motion primitives which represent the elementary compositions of driving skills. Therefore, the purpose of this paper is to segment unlabeled trajectory data into a library of motion primitives. By applying a probabilistic inference based on an iterative Expectation-Maximization algorithm, our method segments the collected trajectories while learning a set of motion primitives represented by the dynamic movement primitives. The proposed method utilizes the mutual dependencies between the segmentation and representation of motion primitives and the driving-specific based initial segmentation. By utilizing this mutual dependency and the initial condition, this paper presents how we can enhance the performance of both the segmentation and the motion primitive library establishment. We also evaluate the applicability of the primitive representation method to imitation learning and motion planning algorithms. The model is trained and validated by using the driving data collected from the Beijing Institute of Technology intelligent vehicle platform. The results show that the proposed approach can find the proper segmentation and establish the motion primitive library simultaneously.

## I. INTRODUCTION

In order to increase the potential for acceptance of intelligent vehicles, the human factors should be regarded as one of the crucial factors in system design [1]–[3]. A commonly accepted concept to achieve such human-like behavior is imitation learning which enables the system to learn from the demonstrations and subsequently reproduce the learned skills to new situations [4]–[6]. Regarding complex driving task as a sequence of motion primitives (MPs) provides two obvious benefits. First, the motion planning tasks will be simplified to desired points setting and the selection of MPs from the library. Second, the learned library of the MPs can also be reused to other driving tasks. These are essential for the motion planning system, as the system can be adapted to different situations by only replacing several point-to-point MPs within the trajectory sequence. A fundamental issue of such methods is acquiring the MPs autonomously without relying on the labeled driving data. In this paper,

we propose a framework for segmenting unlabeled vehicle trajectory data into a library of MPs to solve this problem. Two issues need to be considered in the proposed framework, the segmentation of the observed driving trajectory and the representation of the segmented trajectory primitives.

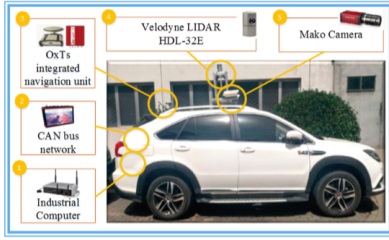
The development of motion planning algorithms is crucial for the intelligent vehicle system. Most of the research groups apply interpolation or graph search to solve the motion planning problem in real implementations [7]. In the method of interpolation, the trajectories are often represented by clothoids [8], Bezier curves [9] and polynomial curves [10]. Graph search is another algorithm which searches the optimal path considering several constraints [11], [12]. Urmson et al. [11] proposed to define the motion as the longitudinal and lateral parameterized functions separately. The linear velocity profile takes the form of a constant profile, linear profile, linear ramp profile or a trapezoidal profile. The lateral part is defined as a second-order spline profile. While the above-mentioned representations are proved to be favorable and effective for optimized trajectory generation, there is no consensus that they are equally applicable to learning algorithms. By using the collected driving data, Schnelle et al. [13] proposed a combined driver model which includes a compensatory transfer function and an anticipatory component based on road geometry to generate the personalized driver’s desired trajectory. However, this approach relies heavily on parameterized driving situations and therefore misses the generality to be chosen as the basic function to establish the motion primitive library.

The representation approach which we used in this paper is based on the dynamical systems and called the Dynamic system movement primitives (DMPs). DMPs have many advantages over other representation methods, such as being robust against perturbations, allowing changing of the final position and time duration without altering the overall shape of the trajectory. Furthermore, DMPs are suitable for imitation learning [14] and reward-driven self-improvement [15], thus have been successfully applied to learn numerous motor skills in robotics [16]. And the finitely varying joint angular values are utilized directly as the parameters of the DMPs in the field of robotics. However, in the driving situation, there are significant changes in the values of the velocity and course angle but limited changes in the relative variation. Considering the specificity of driving situation and differing from the traditional DMPs that widely used in the robotics, the parameters of the proposed MPs are defined by the initial conditions of each motion primitive and control increments within the time duration.

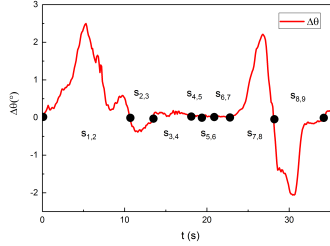
\*This work was supported by the National Natural Science Foundation of China (No.91420203 and No.61703041, and the paper is also funded by International Graduate Exchanging Program of Beijing Institute of Technology

<sup>1</sup> All the authors are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China, 100081 gongjianwei@bit.edu.cn

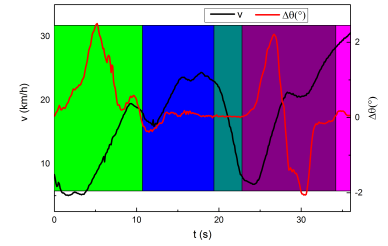
<sup>2</sup> Boyang Wang is also with the Interactive Digital Human group of CNRS-UM LIRMM, UMR5506, Montpellier, France, 34095. wbythink@hotmail.com



(a) A total of 81 hours of driving data are collected by the platform. The path data which are gathered from the integrated navigation unit are utilized in this paper.



(b) Sampled-data of one typical demonstration. The points show the initial segmentation points.



(c) The result of final segmentation and learned MPs. The background color indicates the identified type of MPs.

Fig. 1: The overall flow of our framework. First, the trajectory data are collected by the Beijing Institute of Technology (BIT) intelligent vehicle platform (a). In order to maintain the integrity of the steering process, the initial cutting points are set based on the zero detection of the course deviation (b). According to the initial segmentation, the probabilistic segmentation method is applied based on the representation of the learned MPs. The false segmentation points are merged and the motion primitive library is established (c).

Inferring driver behavior from the driving data is one of the crucial issues in the field of driving assistance system. Some researchers found the algorithms would become more efficient if the complex driving tasks could be decomposed into elemental movements. Li et al. established a driving behavior dataset of thirty-three participants by doing a field operational test around Beijing [17]. The dataset was then used to train the driving style estimation model by concentrating on the transition patterns between typical maneuver primitives. The typical maneuver episodes were segmented manually based on the specific rules designed by the author [18]. Bender et al. [19] developed a two-level unsupervised approach to infer the driver behavior. The naturalistic driving data are automatically segmented in the first step and then assigned to the high-level discrete driving behavior. Taniguchi et al. [20] proposed to model the duration of segments of driving data by using a hierarchical Dirichlet process hidden semi-Markov model. A sequence of driving words is extracted from the data first, and then the driving letters which represent one typical discrete driving behavior are inferred. Inspired by the work of Taniguchi, Hamada et al. [21] extended the work to predict the future continuous operation behaviors using the estimated future state sequence. The states are segmented from the driving data by applying the beta process autoregressive hidden Markov model. Although the unsupervised methods have been successfully applied to segment the driving data based on their features, their focus is on operation behavior data, such as steering angle, brake pressure and accelerator pedal position, they do not concentrate on the particularity of trajectory segmentation. As for the vehicle trajectory analysis, Yao et al. [22] established a lane change trajectory database to analysis the lane change behavior. The trajectories are extracted from the driving data by using the Support Vector Machines based classifier. As a result, the method is limited to extract the specific trajectory based on the labeled data.

In this paper, we apply heuristics to segment the time

series data which is a general way for initial segmentation. Considering the particularity of trajectory segmentation, the zero crossing detection of the observed course deviation is served as the driving-specific heuristic because we prefer to maintain the integrity of the steering process. We apply probabilistic inference to find the correct segments and generate a weighted set of temporary segmented trajectories. The MPs which are learned from the temporary segmented trajectories are then utilized to improve the segmentation result by down-weighting the existed segments which are unlikely to belong to the current motion primitive library. When the iterative Expectation Maximization (EM) algorithm converges, we can get the suitable motion primitive library based on the optimal segmentation points.

The main contributions of this paper are shown as follows.

- Providing one possible method to represent the MPs which is both suitable for the vehicle motion planning task and imitation learning algorithm.
- Implementing a probabilistic segmentation algorithm to solve the segmentation of unlabeled trajectory data and generation of point-point MPs in conjunction.

The remains of this paper are organized as follows. Section III presents the DMPs and probabilistic segmentation methods that we use to solve this problem. Section IV shows the experiment results for the representation and segmentation of MPs. Finally, the conclusions are given in Section V.

## II. PROBLEM STATEMENT

According to a given set of observed vehicle motion trajectories  $\mathcal{O} = \{\mathcal{o}_1, \mathcal{o}_2, \dots, \mathcal{o}_n\}$ , the goal of our work is to learn a set of underlying MPs  $\mathcal{M} = \{\mathcal{m}_1, \mathcal{m}_2, \dots, \mathcal{m}_m\}$  and generate the motion primitive library which can explain  $\mathcal{O}$ .

The BIT intelligent vehicle (see in Fig.1a) was selected as the data-collection platform. The driving situations include intersections, highways and ring roads in Beijing. The op-

erations of the driver contain overtaking, lane keeping, lane changing and other typical driving operations.

The following variables are defined to represent the observed trajectory points and the generated motion primitive points.

- $\mathbf{o}_i(t) = [\Delta\theta(t), v(t)]^T \in \mathbb{R}^{2 \times 1}$  is the definition of one observed vehicle trajectory point at time  $t$ , where  $\Delta\theta(t)$  is the course deviation between time  $t$  and time  $t - 1$ , and  $v(t)$  is the velocity of the trajectory point at time  $t$ .
- $\mathbf{m}_i(t) = [v_{init}, \Delta\theta_m(t), \Delta v_m(t)] \in \mathbb{R}^{3 \times 1}$  is the definition of the one motion primitive point at time  $t$ , where  $v_{init}$  is the initial velocity of the selected motion primitive,  $\Delta\theta_m(t) = \theta_t - \theta_{init}$  and  $\Delta v_m(t) = v_t - v_{init}$  are the generated course deviation and velocity deviation between the initial time and time  $t$  separately. As for the steering, we only care about the relative change, so the initial course angle  $\theta_{init}$  is not included in the definition.

Although the observed trajectories contain the longitudinal and lateral variables, only the lateral part  $\Delta\theta$  is served as the one-dimensional trajectory to determine the segmented points. For the selected trajectory that illustrated in Fig.1, a group of possible cutting points which utilized to indicate the stars and ends of one segmented motion primitive is defined as  $\mathcal{C}$ . We assume that the trajectory is over-segmented by the initial cutting points, i.e., the group of possible cutting points  $\mathcal{C}$  includes both the true positive segments and false positive segments. As a result, each possible subset  $\mathbf{d} \subseteq \mathcal{C}$  has to be regarded as a potential segmentation. Furthermore, the set of segmented trajectories that corresponds to  $\mathbf{d}$  is defined as  $\mathcal{S}$ , where each segmented trajectory  $s_{i,j} \in \mathcal{S}$  is defined by the starts and ends cutting points  $c_i, c_j$  (see in Fig.1b).

The proposed method in this paper is trying to solve the following two challenges simultaneously: Determining the suitable segmented points  $\mathbf{d}^* \subseteq \mathcal{C}$  from the possible cutting points and learning the MPs from the segmented trajectories to generate the library (see in Fig.1c).

### III. LEARNING TO SEGMENT AND REPRESENT MOTION PRIMITIVES

#### A. Initializing the Possible Cutting Points $\mathcal{C}$

As it is unfeasible to consider every time step of the observed trajectories points as the possible cutting points, we should restrict the number of the initial cutting points to an acceptable range. Therefore, some driving-specific heuristics should be used to determine the  $\mathcal{C}$ . Besides, as our method only considers the segmented points included in the  $\mathcal{C}$  and removes the false positive segmented points, the possible cutting points should be a little bit over-segmented. Based on the above considerations, the zero crossing method is applied to initialize the segmented points in this paper, as explained in Fig.1b. Only the lateral variables are chosen to calculate the possible segmented points because we want to guarantee that each steering process is always complete. As a result, the driver's steering behavior and the speed-fitting behavior in the steering process could be completely preserved.

#### B. Representation of Motion Primitive

There are three parameters which are applied to define the MPs. However, the initial velocity  $v_{init}$  is always constant within the time duration of each motion primitive, so only the  $\Delta\theta_m$  and  $\Delta v_m$  need to be represented by the DMPs.

As for the point to point movement, the canonical system is defined as

$$\dot{z}_m = -\tau\alpha_z z_m \quad (1)$$

In order to enable the adaptation for the duration of MPs, the time scaling parameter  $\tau = 1/T$  is defined, where  $T$  is the duration of the MPs. The  $z_m$  is initially set to one and the parameter  $\alpha_z$  is predefined to ensure that the  $z_m$  converges to zero at time  $T$  which is the end time of the chosen motion primitive.

For driving movements, the learned DMPs are defined in the following form

$$\dot{\mathbf{v}} = \tau\alpha_y(\beta_y(\mathbf{g} - \mathbf{y}) - \mathbf{v}) + \tau\eta f(z_m) \quad (2)$$

$$\dot{\mathbf{y}} = \tau\mathbf{v} \quad (3)$$

where  $\mathbf{y} = [\Delta\theta_m, \Delta v_m]$ ,  $\mathbf{v} = [\dot{\Delta\theta}_m, \dot{\Delta v}_m]$ , and the desired final position of the selected motion primitive is  $\mathbf{g} = [\Delta\theta_m(T), \Delta v_m(T)]$ .  $\eta = (\mathbf{g} - \mathbf{y}_0)$  represents the amplitude, where  $\mathbf{y}_0 = [\Delta\theta_m(1), \Delta v_m(1)]$  is the initial position. The pre-defined constants  $\alpha_y$  and  $\beta_y$  are set to ensure the spring-damper system is critically damped.

In order to represent the nonlinear patterns in the randomly selected MPs, the transformation function is defined as

$$f(z_m) = \frac{\sum_{n=1}^N \omega_n \psi_n(z_m) z_m}{\sum_{n=1}^N \psi_n(z_m)} \quad (4)$$

where  $\psi_n(z_m) = \exp(-p_n(z_m - \mu_n)^2)$  is the Gaussian basis functions with  $\mu_n$  represents the center and  $p_n$  represents the bandwidth,  $\omega_n$  is the adjustable parameters for the MPs and  $N$  is the total number of adjustable parameters. The transformation function  $f(z_m)$  changes the output results of the typical spring-damper model and therefore ensures the generation of arbitrarily shaped MPs. The influence of the transformation function  $f(z_m)$  will vanish at the end of the MPs, because the  $z$  converges to zero. This means the generated final position of the movement will converge to the desired final position without any influence of the nonlinear function. That is the foundation for the representation of point-point MPs.

As the predefined parameters  $\alpha_z$ ,  $\alpha_y$  and  $\beta_y$  are always fixed in the representation of motion primitive, the representation parameters for the lateral MPs and longitudinal MPs  $\omega_n = [\omega_\theta(n), \omega_v(n)]$  are needed to be learned from the observed trajectories.

The problem of the learning method is to infer the adjustable parameters  $\omega_\theta, \omega_v$  in order to minimize the errors between the represented and the demonstrated trajectory. Therefore, the problem can be solved directly by a regression algorithm [14]. As the solution process is almost the same between the longitudinal and lateral part, only the detailed

---

**Algorithm 1** Learning of lateral motion primitive
 

---

**Input:**  $M_t = [\Delta\theta_m(t), \Delta\dot{\theta}_m(t), \Delta\ddot{\theta}_m(t)]$ ,  $t \in \{1, \dots, T\}$   
**for** each parameter  $\omega_\theta(n)$  **do**  
 Set predefined parameters  $\alpha_z, \alpha_y, \beta_y, \tau = 1/T, g = \Delta\theta_T$   
  
 Calculate  $z_m(t)$  by integrating  $\dot{z}_m = -\tau\alpha_z z_m$  for all time duration  
 Calculate the deviation value  $f^{dev}(t)$  by using the input data  
 $f^{dev}(t) = \Delta\ddot{\theta}_m(t)/\tau^2 - \alpha_y(\beta_y(g - \Delta\theta_m(t)) - \Delta\dot{\theta}_m(t))$   
  
 Calculate the Gaussian basis functions  
 $\psi_t^n(z_m(t)) = \exp(-p_n(z_m(t) - \mu_n)^2)$   
 Set matrices  $\mathbf{z} = [z_m(1), \dots, z_m(T)]^T$ ,  $\boldsymbol{\psi} = \text{diag}(\psi_1^n, \dots, \psi_T^n)$  and  $\mathbf{f}^{dev} = [f^{dev}(1), \dots, f^{dev}(T)]^T$   
  
 Compute  $\omega_\theta(n)$  by locally weighted regression  
 $\omega_\theta(n) = (\mathbf{z}^T \boldsymbol{\psi} \mathbf{z})^{-1} \mathbf{z}^T \boldsymbol{\psi} \mathbf{f}^{dev}$   
**end for**

---

method for learning lateral MPs from the demonstrated trajectory is shown in Algorithm 1.

By using the Algorithm 1 we can calculate the  $\omega_\theta, \omega_v$  for every segmented MPs, then the representation parameter set and adjustment parameter set for each MPs are  $\phi_m = [v_{init}, \omega_\theta, \omega_v]$  and  $\gamma_m = [g, T]$  separately.

### C. Probabilistic Segmentation Method

Each motion primitive  $m \in \mathcal{M}$  is represented by the parameter set  $\phi_m$ . Therefore, the library of MPs is defined as a mixture of every single motion primitive, in order to calculate the probability of a selected segmented trajectory given the entire library.

$$p(\mathbf{s}|\Phi) = \sum_{m=1}^M \lambda_m p(\mathbf{s}|\phi_m) \quad (5)$$

$$\Phi = \{(\lambda_1, \phi_1), \dots, (\lambda_M, \phi_M)\} \quad (6)$$

where  $\lambda_m$  is the mixture coefficient for each motion primitive in the library.

We assume that each observed trajectory  $\mathbf{o} \in \mathcal{O}$  is represented by a parameterized model.

$$\mathbf{o} \sim p(\mathbf{o}|\Phi, \mathbf{S}^*) = \prod_{s \in \mathbf{S}^*} p(\mathbf{s}|\Phi) \quad (7)$$

The latent variable  $\mathbf{S}^*$  can be integrated out by using Eq.8.

$$p(\mathbf{o}|\Phi) = \sum_{\mathbf{S} \in \mathcal{D}} p(\mathbf{S}) \prod_{s \in \mathbf{S}} p(\mathbf{s}|\Phi) \quad (8)$$

The most likely parameter model  $\Phi$  of motion primitive library is computed by maximizing the likelihood.

$$\Phi^* = \arg \max_{\Phi} \sum_{\mathbf{o} \in \mathcal{O}} \log p(\mathbf{o}|\Phi) \quad (9)$$

We apply the EM algorithm to optimize this likelihood and the auxiliary function is defined as

$$\Phi = \arg \max_{\Phi} Q(\Phi, \Phi') \quad (10)$$

$$Q(\Phi, \Phi') = \sum_{\mathbf{o} \in \mathcal{O}} \sum_{\mathbf{S} \in \mathcal{D}} p(\mathbf{S}|\mathbf{o}, \Phi') \log(p_{\mathbf{S}} p(\mathbf{o}|\Phi \mathbf{S})) \quad (11)$$

where  $\Phi'$  is defined as model parameter set which is calculated from the previous iteration. The  $\alpha_s = \sum_{\mathbf{S} \in \mathcal{D}} p(\mathbf{S}|\mathbf{o}, \Phi')$  indicates the probability of each segment to be the correct segmentation. The prior  $p_s$  is defined as a product of priors over segments.

$$p_s = p_c \prod_{s \in \mathbf{S}} p(c_s) \quad (12)$$

$$p(c_s) = (1 - p_c)^{c_s} p_c \quad (13)$$

where  $c_s$  is the total number of cutting points that the possible segmented trajectory  $s$  spans over. The parameter  $0 < p_c < 1$  indicates how suitable it is that the cutting points which determine the segmented trajectory  $s$  are the true positive. When  $p_c < 0.5$  indicates that longer segmented trajectory is preferred. On the contrary, shorter segments are preferable.

By using the function defined in Eq.11, the weighting parameter  $\alpha_s$  is computed from the E-step and the Gaussian Mixture Models (GMMs) based parameter set  $\Phi'$  is estimated in the M-Step. Once the algorithm converges, we can simultaneously get the correct segmented points and the learned GMMs to represent the parameter set  $\Phi$  of motion primitive library.

### D. EM-GMM Segmentation Method

The EM-GMM segmentation method is selected as the baseline method to compare with the probabilistic segmentation method. The initial possible cutting points and the DMP are not applied in this method.

In the training process of the EM-GMM segmentation method, the feature dataset  $\mathbf{x}_p(t)$  is chosen as

$$\mathbf{x}_p(t) = \mathbf{o}_i(t) = [\Delta\theta(t), v(t)] \quad (14)$$

The GMM is represented as

$$G(\mathbf{x}; \boldsymbol{\beta}) = \sum_{i=1}^K p_i g_i(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (15)$$

$$g_i(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_i|}} e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)} \quad (16)$$

where  $\mathbf{x}$  is a set of d-dimension sequence,  $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^n$  with  $x_i \in \mathbb{R}^{d \times 1}$ ,  $\boldsymbol{\mu}_i \in \mathbb{R}^{d \times 1}$  and  $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d \times d}$  are mean vector and covariance vector of a single Gaussian Model,  $p_i$  is the weight coefficient of a single Gaussian Model and  $\sum_{i=1}^n p_i = 1$ ,  $\boldsymbol{\beta} = \{\boldsymbol{\beta}_i\}_{i=1}^n$  with  $\boldsymbol{\beta}_i = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, p_i\}$ . Maximum Likelihood Estimation of the model parameters is achieved iteratively using the Expectation-Maximization (EM) algorithm.

After the model has been trained by applying the chosen path features, the path point sequences can be labeled by the trained model. Therefore, the trajectories are segmented by the learned labels, and the path primitives are based on the  $\mathbf{x}_p(t)$  not the parameter set  $\Phi$  which is used in the probabilistic segmentation method.

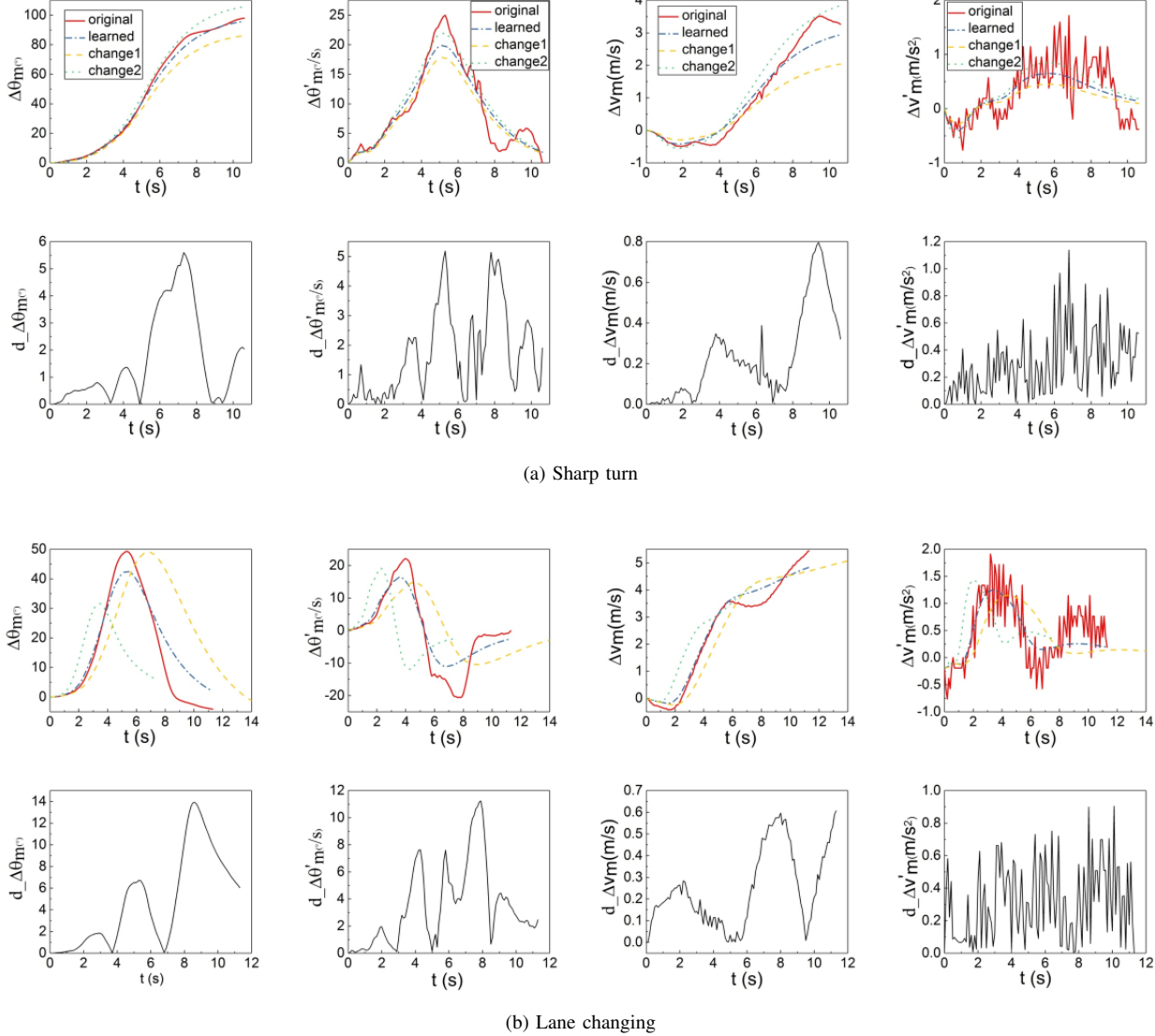


Fig. 2: This figure illustrates the accuracy of our MPs representation method and the adaptation of the learned MPs to the change of final goal position and time duration which are essential for both the imitation learning and motion planning. Two typical driving operations were selected to evaluate the performance of representation method. In each situation, the goal  $\Delta\theta_m$ ,  $\Delta\dot{\theta}_m$ ,  $\Delta v_m$ ,  $\Delta\dot{v}_m$  and the corresponding deviations were presented in order. Situation a concentrated on the goal value change of  $\Delta\theta_m(T)$ ,  $\Delta v_m(T)$  and situation b demonstrated the change in time duration  $T$ . The learned parameter set  $\phi_m$  remains unchanged in the adjustment. It could be found that the overall shape of the trajectory is not changed during the adjustment.

## IV. EXPERIMENTS AND RESULTS

### A. Data Collection System

The proposed method in this paper mainly focuses on the drivers driving skills and does not involve the comparison of the multi-driver driving styles. As a result, only one drivers driving data was selected from the 81 hours driving data to train the model in this paper.

The trajectory dataset which was mainly used in this paper was collected by the equipped OxTs integrated navigation unit. Course angle, horizontal velocity and timestamp of the collection were provided by this sensor.

### B. Evaluation of Typical Motion Primitives

There are a total of 26 basic MPs in the library. However, it was meaningless to demonstrate every motion primitive in the library. As a result, only two typical MPs were selected to evaluate the performance of our representation method. The two selected typical motion primitives were sharp turn and lane changing motion primitive. The reasons we chose these two motion primitives is that the two common steering processes contain typical longitudinal-lateral cooperation relationship which is the crucial factor that the learned motion primitives would like to represent.

In order to evaluate the accuracy of our representation method, the position and velocity deviations were chosen as the performance indexes. The position deviation  $d_{pos}$  can be described as

$$d_{pos} = |y_d - y_l| \quad (17)$$

The velocity deviation  $d_{vel}$  can be calculated by

$$d_{vel} = |v_d - v_l| \quad (18)$$

where  $y_{d/l}$  and  $v_{d/l}$  are the motion primitive data of the position and velocity respectively, the subscript  $d$  and  $l$  represent the demonstrated and learned data respectively.

We also presented the adaptability to the change of goal position and time duration (see in Fig.2).

### C. Comparison of the Trajectory Segmentation

In order to demonstrate the benefits of optimizing both the trajectory segmentation and the motion primitive library iteratively, we applied the EM-GMM method as a baseline. The baseline method clusters the dataset  $\mathcal{O}$  directly in order to get the identified trajectory type and segment the trajectory based on the identified type. The segmentation comparison results based on these two methods were shown in Fig.3.

The performance of segmentation method was evaluated by the number of active segmented points  $n_2$  (see in Table.I), and less segmentation points mean more compact representation of the observed trajectory.

Type	$n_1$	$n_2(\text{Probs})$	$n_2(\text{EM-GMM})$
1	37	14	42
2	82	8	94

TABLE I: The segmentation results of probabilistic segmentation (Probs) results and EM-GMM method. The type 1 indicates the driving situation of low-speed with sharp turns, and type 2 indicates the driving situation of high-speed with direction corrections.  $n_1$  is the initial cutting points of the selected trajectory, and  $n_2$  is the number of active segmented points after the learning process.

### D. Discussion

The benefit of our approach becomes quite obvious when comparing the probabilistic segmentation approach with the reference EM-GMM method, shown in Fig.3. Despite achieving comparable segmentation results, the proposed method can achieve more meaningful segment results which correspond to the realistic driving situations compared with the baseline method. The reason is that the probabilistic segmentation is based on the initial cutting points which are determined by the prior knowledge of the motion primitive settings. The purpose of the proposed method is to merge the positive false cutting points instead of directly finding the positive true segment points. However, the EM-GMM method or other similar learning methods which automatically extract MPs from driving data without any prior knowledge are suitable for many applications, but they may

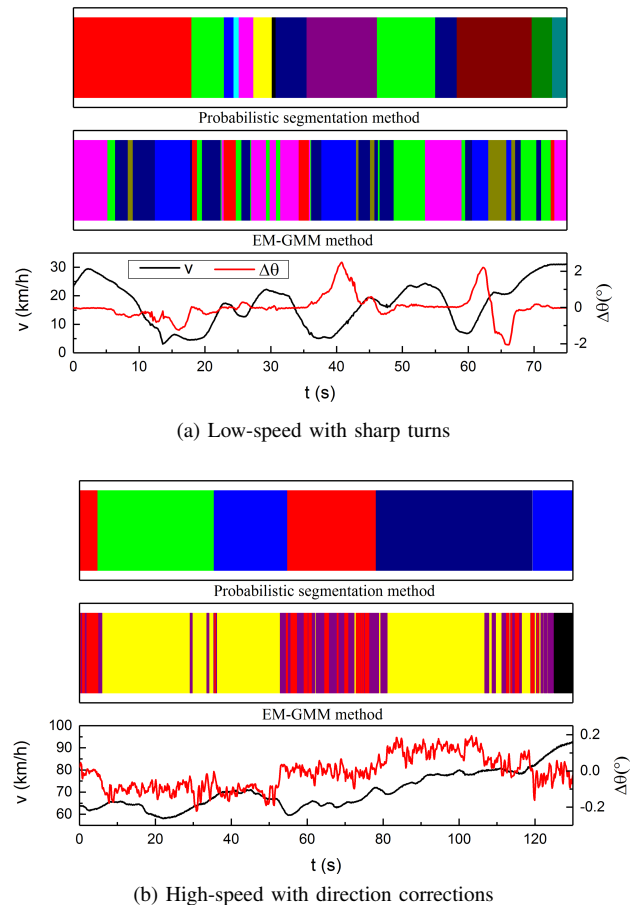


Fig. 3: The segmentation results of two typical driving situations. In each subgraph, the segmentation result of the probabilistic segmentation method, the segmentation result of the EM-GMM method and the observed path data were presented in order. The EM-GMM was chosen as the representative algorithm of unsupervised learning approaches which can automatically extract MPs from driving data without any initial heuristics. Different colors within each segmentation result illustrated different types of MPs, but there was no association between colors in different result graphs. Some of the segments of the EM-GMM method were too short to be seen in the plot.

not be the proper methods to segment the trajectory for the motion planning applications. As for the MPs which are used for the motion planning tasks, some heuristics can and must be made previously for meaningful segmentation and reusing of segmented MPs.

Besides, as our method takes advantage of the mutual dependencies between the segmentation and representation, so the representation method is another crucial factor. As shown in Fig.2, the proposed representation method can be adapted to the change of final goal position and time duration with the same representation parameter set  $\phi_m$ . It is essential for both the representation and reusing of the learned motion primitive. As for the representation, the trajectory can be represented by the same set of parameters

as long as the basic shape of the trajectory is the same. This will greatly reduce the number of possible primitive types and enhance the effectiveness of the segmentation algorithm. As for reusing of MPs to motion planning application, we can achieve different target points with the same overall shape of the trajectory by only changing the adjustment parameter set  $\gamma_m$ . The representation parameter set  $\phi_m$  which is served as the identification of different shape types remains unchanged during the adaptation. This property is the basis of how DMPs can be used as a representation method for the MPs library. Furthermore, the representation method is also robustness to sudden perturbations and can resist fluctuations near zero. However, it is unable to represent emergency driving behaviors which contain sudden changes in the values of the trajectory. As a result, all the emergency driving situations are removed from the training dataset manually with the help of collected camera data.

## V. CONCLUSION

In this paper, we proposed a new framework which can segment the unlabeled trajectories and learn a motion primitive library in conjunction. The established motion primitive library is used to infer the true positive segment points from the possible initial cutting points which are chosen based on the zero detection of the observed course deviation. Then the inferred segmentations, in turn, help to optimize the MPs in the library by utilizing the mutual dependency between the segmentation and representation. We evaluated the representation accuracy and adaptation of the learned MPs by using the driving data collected from the BIT intelligent vehicle platform, and we also compare our segmentation method with one of the typical unsupervised learning approaches. We found that our approach is able to learn more compact and meaningful MPs compared with the unsupervised learning method without any driving-specific heuristics, and the learned MPs which can adapt to the change of goal position and time duration are suitable for both the imitation learning and motion planning. The proposed method can be applied to learn the human driving skills from the collected driving data in a more efficient way, and the learned MPs library can be combined with the traditional motion planning method to generate personalized motion pattern.

In future work, we aim at adjusting the representation method of the MPs to make it more suitable for driving tasks, combining the proposed approach with some stable motion planning methods and optimizing the MPs online.

## REFERENCES

- [1] Clara Marina Martinez, Mira Heucke, Fei-Yue Wang, Bo Gao, and Dongpu Cao. Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):666–676, 2018.
- [2] Scott Schnelle, Junmin Wang, Richard Jagacinski, and Hai-jun Su. A feedforward and feedback integrated lateral and longitudinal driver model for personalized advanced driver assistance systems. *Mechatronics*, 50:177–188, 2018.
- [3] Scott Schnelle, Junmin Wang, Hai-Jun Su, and Richard Jagacinski. A personalizable driver steering model capable of predicting driver behaviors in vehicle collision avoidance maneuvers. *IEEE Transactions on Human-Machine Systems*, 47(5):625–635, 2017.

- [4] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.
- [5] Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.
- [6] Rudolf Lioutikov, Gerhard Neumann, Guilherme Maeda, and Jan Peters. Probabilistic segmentation applied to an assembly task. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 533–540. IEEE, 2015.
- [7] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2016.
- [8] Helene Vorobieva, Sebastien Glaser, Nicoleta Minoiu-Enache, and Said Mammar. Automatic parallel parking with geometric continuous-curvature path planning. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 465–471. IEEE, 2014.
- [9] Plamen Petrov and Fawzi Nashashibi. Modeling and nonlinear adaptive control for autonomous vehicle overtaking. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1643–1656, 2014.
- [10] Joshue Perez Rastelli, Ray Lattarulo, and Fawzi Nashashibi. Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 510–515. IEEE, 2014.
- [11] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
- [12] Thomas M Howard and Alonzo Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *The International Journal of Robotics Research*, 26(2):141–166, 2007.
- [13] Scott Schnelle, Junmin Wang, Haijun Su, and Richard Jagacinski. A driver steering model with personalized desired path generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):111–120, 2017.
- [14] Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1398–1403. IEEE, 2002.
- [15] Jens Kober and Jan R Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856, 2009.
- [16] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.
- [17] G. Li, S. Eben Li, and B. Cheng. Field operational test of advanced driver assistance systems in typical chinese road conditions: The influence of driver gender, age and aggression. *International Journal of Automotive Technology*, 16(5):739–750, 2015.
- [18] Guofa Li, Shengbo Eben Li, Bo Cheng, and Paul Green. Estimation of driving style in naturalistic highway traffic using maneuver transition probabilities. *Transportation Research Part C*, 74:113–125, 2017.
- [19] Asher Bender, Gabriel Agamennoni, James R Ward, Stewart Worrall, and Eduardo M Nebot. An unsupervised approach for inferring driver behavior from naturalistic driving data. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3325–3336, 2015.
- [20] Tadahiro Taniguchi, Shogo Nagasaka, Kentarou Hitomi, Naiwala P Chandrasiri, Takashi Bando, and Kazuhito Takenaka. Sequence prediction of driving behavior using double articulation analyzer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(9):1300–1313, 2016.
- [21] Ryunosuke Hamada, Takatomi Kubo, Kazushi Ikeda, Zujie Zhang, Tomohiro Shibata, Takashi Bando, Kentarou Hitomi, and Masumi Egawa. Modeling and prediction of driving behaviors using a nonparametric bayesian method with ar models. *IEEE Transactions on Intelligent Vehicles*, 1(2):131–138, 2016.
- [22] Wen Yao, Qiqi Zeng, Yuping Lin, Donghao Xu, Huijing Zhao, Franck Guillemard, Stephane Geronimi, and Francois Aioun. On-road vehicle trajectory collection and scene-based lane change analysis: Part ii. *IEEE Transactions on Intelligent Transportation Systems*, 18(1):206–220, 2017.