Driving Assistant Companion with Voice Interface using Long Short-Term Memory Networks

Jehyun Park, Hojoon Son, Jisuk Lee, and Jongeun Choi

***Abstract*—In this paper, we propose a driving assistant companion system that provides drivers with useful information using a long short-term memory (LSTM) network in a data-driven fashion. Our system can be viewed as an advanced driver-assistance system (ADAS) for faster learning and better driving. The assistant companion predicts upcoming events on the road by using real-time sensory measurements from range finding sensors and provides informative narrations to enhance learnability and driving performance. In contrast to a conventional navigation system, our system predicts events from the online-stream of sensory measurements without resorting to priors and map information. To demonstrate the effectiveness of the proposed system, we implemented our system on The Open Racing Car Simulator (TORCS) and conducted an experimental study with 16 human drivers. Experimental results show that our system enhanced learnability, driving performance, and reliability.**

***Index Terms*—ADAS, Driving assistant companion, Learnability, TORCS, LSTM, Voice assistance, Driving performance, Reliability.**

I. INTRODUCTION

Recently, car makers and information technology companies such as Audi AG, NVIDIA Corp., Google Inc., and Intel Corp. collaboratively strive to develop better advanced driver assistance systems (ADAS) and autonomous driving [1], [2]. ADAS are developed to assist a driver for convenience, safety, and performance by fusing various sensory measurements from LiDAR, radar, vision sensors, etc. [3]. For example, they may recognize road signs from the vision data, such as speed limits and warnings that can be overlooked while driving to improve safety [4], [5]. The benefits of ADAS implementations are potentially considerable because of a significant decrease in human suffering, economical cost and pollution [6]. There have been various studies showing that ADAS improves driving performance. F. Biral *et al.* [7] combined user's preferred driving style and safety margins into an ADAS's module for optimal reference maneuver computation. For safety on driving, there are physical and visual requirements such as keeping your hands on the steering wheel and consistently looking ahead [8], [9]. Real-time driving assistance through a voice interface is applicable under such constraints in dynamically changing environments. To this end, smart voice assistant systems such as Alexa have begun to be applied to ADAS by several car makers [10]. Alexa is an intelligent personal assistant which can provide various services with a voice user interface alone [11]. A voice assistance such as 'Release throttle', 'Brake smoothly', or 'Brake sharply', depending on the vehicle speed, the distance between the vehicle and the traffic signal, etc., is shown to save up fuel consumption up to 25% [12]. In addition, cognitive and sensory declines in the elderly people might impact on the ability to drive [13], where ADAS with a voice interface might be useful and effective.

In this paper, we consider a problem of building an ADAS with a voice interface for drivers to learn and perform better on a challenging driving task. We propose a driving assistant companion system as an ADAS that enables to improve driver's learnability and performance by providing the driver with useful information through a voice interface in real-time. The connectedness through a voice interface will provide a psychological stability [14], which might provide a companionship. The companionship plays an important role during the long-term use of intelligent systems with intimacy.

We propose to use a long short-term memory (LSTM) neural network in fusing environmental sensory measurements and in generating appropriate responses in real-time. The LSTM networks have been shown to be successful to analyze time-series data for detecting anomalies in ECG signals [15], [16] and automobile control network data [17].

The contributions of the paper are as follows. Firstly, to demonstrate the capability of our assistant driving system, we implemented our system using The Open Racing Car Simulator (TORCS). For practicality, we choose 19 range finding sensors (in TORCS), each of which returns the distance between the track edge and the ego-vehicle within a range of 200 meters. Note that our system uses only range finding sensor measurements, therefore it is applicable in environments without priors and map information. Next, the LSTM neural network was used to predict particular driving situations. The LSTM structure is designed to receive the sensory measurements as an input value to classify situations of interest. To compute the prediction, we trained the LSTM model with the sensory measurements in manually labeled situations to obtain learned weights. While driving, our trained model successfully predicts events in real-time in a dynamic environment, and the classification results are reported in this paper. Most importantly, to evaluate our approach in an effective manner, we designed an experiment study with a challenging task using a control delay and a difficult track containing multiple corners. We used Mann-Whitney U-test to show the increased performance and reliability as compared to

This work was supported by the Ministry of Trade, Industry and Energy (MOTIE) and the Korea Institute for the Advancement of Technology (KIAT) (N0002385, 2017).

Jehyun Park, Hojoon Son, Jisuk Lee and Jongeun Choi are with the School of Mechanical Engineering, Yonsei University, 50 Yonsei Ro, Seodaemun Gu, Seoul 03722, South Korea (e-mail: jhyunpark@yonsei.ac.kr; hojoonson@yonsei.ac.kr; jslee@motion-device.com; jongeunchoi@yonsei.ac.kr).

Jongeun Choi is the corresponding author (e-mail: jongeunchoi@yonsei.ac.kr).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2861739, IEEE Transactions on Industrial Informatics

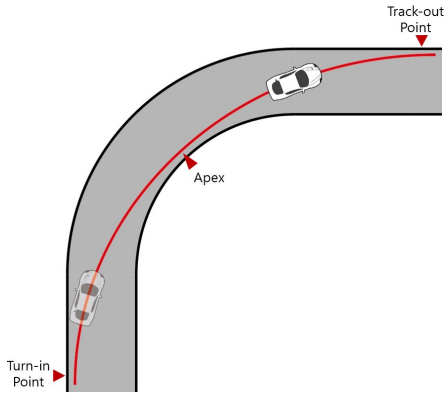

Fig. 1. Optimal path (record line).

ones from the control group without our system. The paper is organized as follows. In Section 2, we explain the experimental design and how the driving assistant companion system assists a novice driver. Section 3 presents how we build our driving assistant companion system using learned weights through the LSTM network to generate prediction. Section 4 presents the experiment design and its results are shown in Section 5. We discuss the results in Section 6 and concluding remarks are shown in Section 7.

II. METHOD OF ASSISTANCE

A. Challenging Driving Task

We demonstrate our approach using TORCS. Due to its openness, modularity, and extensibility, TORCS has been adopted as a framework for many research projects [18], [19], [20]. While driving on TORCS, our system extracts sensory measurements and deliver appropriate responses to a driver through a speaker. In order to test capability of our system effectively, we designed a challenging driving task. Firstly, we designed a testing track from the TORCS-based driving game with many corners that requires skillful driving such that the performance depends on how well a driver passes multiple corners. Next, we inserted a control delay (0.2 sec) between the control input of the driver and the game system to make the task challenging as shown in Fig. 3. Delayed driving control responses are typical processes that will impair driving performance as often caused by alcohol and drugs [21]. Furthermore, cognitive and sensory declines in the elderly population might be modeled by adding a delay in the feedback loop [13].

B. Classified Situational Events and the Voice Assistance

In this section, we explain how we classify situational events and how we guide the optimal path via a voice interface. To successfully cope with the corners, the optimal path is one that minimizes the time spent through the corner to maximize the overall speed. The optimal path is that the driver starts turning at the turn-in point (Fig. 1) and comes inward at the middle of the curve to pass the apex (Fig. 1). The driver then exits to the outside of the curve, at the track-out point (Fig. 1). As you follow the optimal path [22], you can maximize the radius

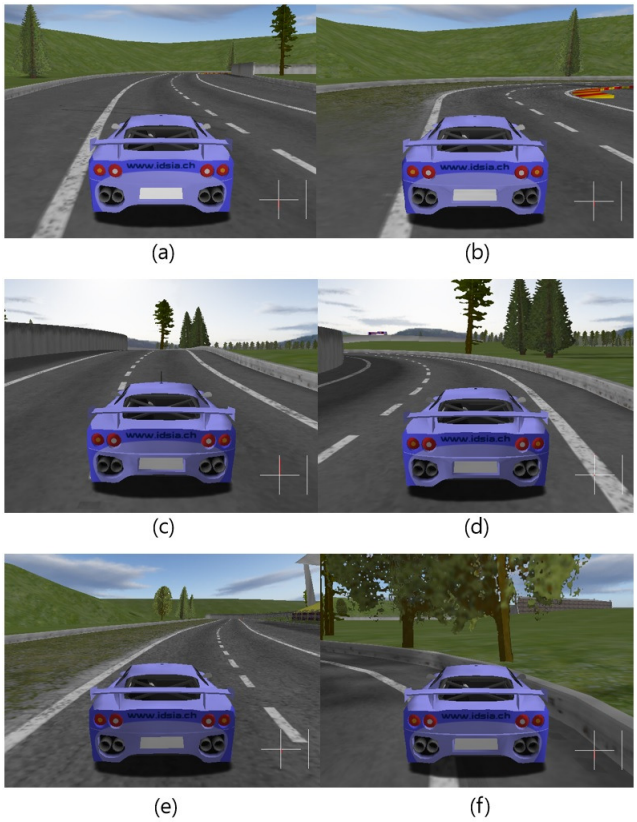

Fig. 2. Snapshots of different situational events. The corresponding narrations are given by (a) Stay left (b) Turn right (c) Stay right (d) Turn left (e) Accelerate (f) Be careful.

of the curvature to pass the curve at the maximally possible speed. To guide the optimal path, when the companion predicts a right hand curve is coming, it speaks out 'Stay left' (Fig. 2-(a)) to pass the turn-in point. If a driver is about to be at the turn-in point of the right hand curve, it speaks out 'Turn right' (Fig. 2-(b)) to approach the apex. When the companion predicts a left hand curve, the appropriate voice assistance can be applied in a symmetrical way. In addition, if a driver is on a straight track, the companion speaks out 'Accelerate' (Fig. 2-(e)). Finally, when the driver is on the verge of crash, the companion speaks out 'Be careful' (Fig. 2-(f)). Therefore these voice information will induce the player's learnability and performance towards optimality.

III. MACHINE LEARNING FOR CLASSIFICATION

To train our machine learning algorithm for classifying the driving situational events, we collected sensory measurement data under 6 different classes as in Fig. 2. Table I shows the name of each class, which is defined according to the narration for each situation. With the collected situational data and the corresponding classes, we trained the LSTM network (Fig. 3). The LSTM network is powerful to deal with the time-series data in a very fast dynamic environment such as racing games [23]. The LSTM network training is performed in offline (Fig. 3). Finally, we build an algorithm to classify the current situation with the given online sensory measurements in TORCS. In order to achieve the optimal performance, it is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2861739, IEEE Transactions on Industrial Informatics

TABLE I
CLASS LABEL DEFINITION

Narration for situation	Class
Stay left	SL
Turn right	TR
Stay right	SR
Turn left	TL
Accelerate	AC
Be careful	BC

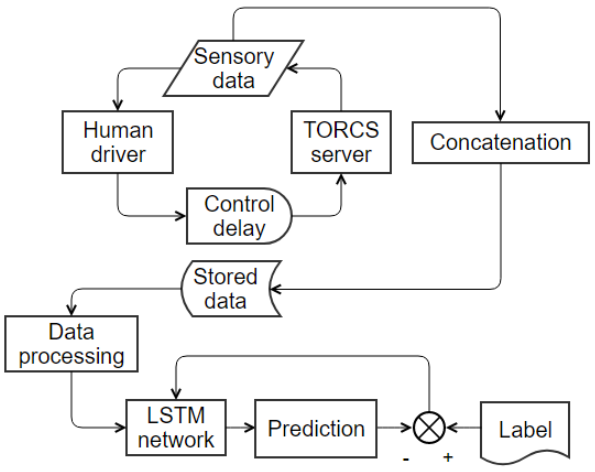

Fig. 3. The flowchart of the training procedure.

important not only to classify exactly what the current situation is, but also to foresee what will happen in the future. The class SL and SR foresee that the curved track is coming. In other words, the classes SL and SR will be trained by the time-shifted, time-series data, which are initially used for the classes TR and TL.

A. LSTM Model used for Classification

In this section, we explain the LSTM model that will be trained to classify and predict the situational events. The LSTM network solves the vanishing gradient problem, which pertains to the loss of information with time due to decaying gradient values [16]. A conventional LSTM network has three layers, viz. a hidden layer, an LSTM layer, and an output layer [16]. Hidden and output layers are non-recurrent and have multiple units. The LSTM layer is recurrent and has its memory blocks which includes multiple cells [24]. Our LSTM network has a hidden layer, an output layer, and two LSTM layers, which is called a deep LSTM network [16].

B. Model Selection and Network Structure

With the collected situational data and the corresponding classes, we tested the LSTM network with the different number of LSTM layers for model validation. Based on 200 iterations, the average accuracy using 10-fold cross-validation was 91.85% for one layer, 98.27% for two, 98.35% for three, and 96.10% for four layers. We decided that it is more

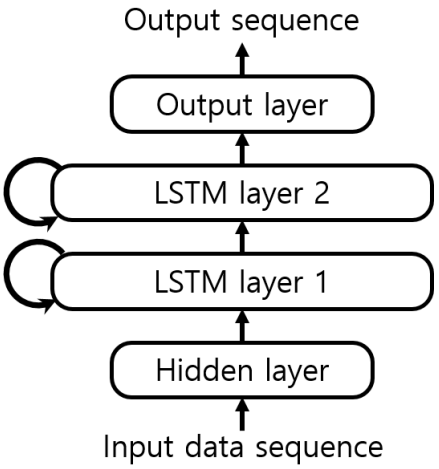

Fig. 4. The LSTM network model used to classify and predict the situational events.

appropriate to use two LSTM layers by the law of parsimony (i.e., Occam's razor) because we already had enough accuracy with two layers. Furthermore, the number of nodes in the input and output layers are determined with appropriate dimensions based on the size of the input data and the number of classes. The number of nodes in the hidden layer takes into account the number of the input nodes and is determined to minimize the loss function. Finally, the learning rate is tuned to be a bit low in order to stabilize the learning process (Fig. 6).

In our model, the input data sequence is transformed by a hidden layer that has 32 units. The data from the hidden layer are fed into the LSTM layers. Each LSTM layer has five memory blocks, which include 32 cells each. The data from the LSTM layers are transformed by an output layer that has 6 units. In summary, Fig. 4 shows our LSTM network model. In what follows, we explain the cell structure of the LSTM layers. A cell of the LSTM layers is a chained structure. There are four parts. The first part is to decide what information to forget from the cell state.

$$f_t = \sigma(W_f h_{t-1} + W_f x_t + b_f), \tag{1}$$

where f_t is the forget gate activation vector at time t, the number of the hidden nodes is denoted by p, σ is a logistic sigmoid function, W_f is the matrix of weights on the input from the forget gate, h_t is the output at time t, x_t is the input sequence, and b_f is the forget gate bias vector. The second part is to decide what new information we store in the cell state c.

$$i_t = \sigma(W_i h_{t-1} + W_i x_t] + b_i), \tag{2}$$

$$\tilde{c_t} = \tanh(W_c h_{t-1} + W_c x_t + b_c), \tag{3}$$

where i_t is the input gate activation vector at time t, W_i is the matrix of weights to the input from the input gate, b_i is the input gate bias vector, W_c is the matrix of weights to the input from the cell, $\tilde{c_t}$ is the vector of new candidate values, and b_c is the cell state bias vector. The third part is to update

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2861739, IEEE Transactions on Industrial Informatics

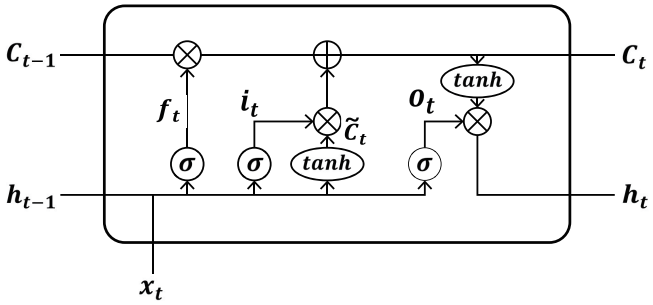

Fig. 5. A chain structured LSTM cell.

the cell value from the old cell state value, i.e., from c_{t-1} into the new cell state c_t at time t.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{4}$$

where $\odot$ is the element-wise product operator between vectors. Finally, we define the output gate o_t and the output h_t at time t.

$$o_t = \sigma(W_o h_{t-1} + W_o x_t + b_o), \tag{5}$$

$$h_t = o_t \odot \tanh(c_t), \tag{6}$$

o_t is the output gate activation vector, W_o is the matrix of weights on the input from the output gate, and b_o is the output gate bias vector. As shown in Fig. 4, a training data matrix $D \in \mathbb{R}^{m\times n}$ is multiplied by the hidden weight matrix $W_{hidden} \in \mathbb{R}^{n\times o}$, where m, n, and o denote the numbers of time-step, sensory measurements, and nodes in the hidden layer, which are 5, 19, 32 in our model, respectively. A constant bias is added to it in a element-wise manner and the resulted matrix is fed into a network of two LSTM layers (or cells) as shown in Fig. 4. The output matrix $O_{LSTM} \in \mathbb{R}^{1\times o}$ of the network of two LSTM layers is multiplied by the output weight matrix $W_{output} \in \mathbb{R}^{o\times p}$ and with an added bias vector b_{output}, we obtain the score matrix

$$s := O_{LSTM} W_{output} + b_{output} \in \mathbb{R}^p, \tag{7}$$

where p denotes the number of class, which is 6 in our model. Each value of s is converted to the probability using the softmax function [25]. The softmax transformed value is given by

$$p_i := \frac{e^{s_i}}{\sum e^{s_i}}, \quad i \in \{1, 2, \cdots, p\}. \tag{8}$$

C. Data Collection for Machine Learning

For our classification problem, we need to collect and store the enough data for each situational event so that the LSTM network can learn. Once every 0.2 seconds, we collect the sensory measurements to create a row vector $x_\ell \in \mathbb{R}^{1\times n}$ at each time step ℓ, where $n = 19$. x_ℓ contains n range finding sensor measurements. The i-th element is the distance between the track edge and the car measured by the i-th sensor. Sensors are distributed 10 degrees apart each other in front of the vehicle. We then cumulate them into X_ℓ by

$$X_\ell := X_{\ell-1} \| x_\ell \in \mathbb{R}^{\ell\times n}, \quad \ell = 1, 2, \cdots, \tag{9}$$

by concatenating x_ℓ to $X_{\ell-1}$, where the concatenation operator is denoted by $\|$ and defined by $X\|Y = [X^T Y^T]^T$.

In order to collect the data during a specific time interval for each situation, we assigned 6 keyboard keys for each situation using a Python module that detects a keyboard input. Keyboard keys act like on/off switches. The data concatenating cycle runs only when the keyboard input is initialized. Pressing the same key terminates the data concatenating cycle and terminates the data collection. In addition, other keys operate individually, so the data are collected separately for each class.

If a specific situation occurs while driving the car, the key assigned to the situation is pressed to start the data collection. After a certain period of time when the situation is over, the same key is pressed to stop collecting the data.

By repeating this process, we collected the training and test data sets for the corresponding classes SL, TR, SR, TL, AC, and BC, (with the dimensions of $\ell \times n$, i.e., $3430 \times 19, 2565 \times 19, 1850 \times 19, 1360 \times 19, 1970 \times 19$, and 1535×19), respectively. To better classify the situational event, the LSTM network needs the time-series data of several time steps rather than at each sampling time. To train and predict the event with a time-windowed data during the last 1 sec., all the collected measurements from the range finders are time-windowed into a data matrix $D_\ell \in \mathbb{R}^{m\times n}$ ($m = 5, n = 19$) at time ℓ.

D. Training for Classification Learning

Each of the data matrices is labeled with one of p true class vectors

$$c^{(1)}_{1\times p}, \cdots, c^{(p)}_{1\times p} \in \mathbb{R}^{1\times p}, \tag{10}$$

where $p = 6$ according to the situational events, such that $c^{(j)}_{1\times p}$ has all zeros except 1 at the j-th entry for all $j \in \{1, 2, \cdots, p\}$. Therefore we have that

$$c^{(j)}_i = \delta_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j, \end{cases} \tag{11}$$

where δ_{ij} is the Kronecker delta. The weight matrices and biases (together denoted by W) in the LSTM network (Fig. 4) are updated in every iteration using a gradient descent method (i.e., back propagation) to reduce the total loss function that is the sum of the cross entropy function and the L_2 loss function [26].

$$W_{k+1} = W_k - \lambda_k \nabla(H(W_k) + \lambda_r \|W_k\|_2), \tag{12}$$

where λ_k, λ_r, and H are the learning rate, the regularization coefficient, and the cross-entropy function respectively. The cross entropy function is given by

$$\begin{aligned} H(p_i, c^{(j)}_i) &:= -\sum_i c^{(j)}_i \log(p_i), \quad i \in \{1, 2, \cdots, p\}, \\ &= -\sum_i \delta_{ij} \log(p_i), \quad i \in \{1, 2, \cdots, p\}. \end{aligned} \tag{13}$$

Figure 6 shows the total training loss over 10,000 iterations and the total loss is reduced as the number of iterations increases. The total training loss at the last iteration is 0.0874. After the data-driven learning with the LSTM network, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2861739, IEEE Transactions on Industrial Informatics

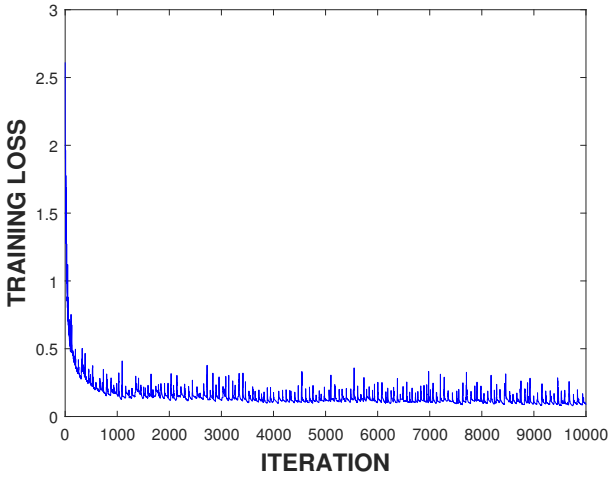

Fig. 6. Training loss vs. iterations.

final weight matrices and biases are used for classifying the situational events during the game in real-time.

In order to provide a better service, we adjusted the timing as well as classifying the situation. The companion speaks once at the beginning so that the driver in the same situation is not repeatedly reminded until the driver encounters another situation. If our system confidently predicts a situation, it speaks only once until another situation is confidently predicted. In addition, we used a specific value as a threshold to provide voice services only when the probability for each situation exceeds that threshold. The threshold is set such that the driver receives more accurate and reliable voice services. Therefore, the companion does not speak unless the threshold exceeds the certain value. In terms of the computation time, it takes 72.21 seconds to perform 1,000 iterations offline training and 0.24 seconds to predict the situation online once the sensory measurements come in. Details of the workstation used in this research are described in Section 4.

IV. EXPERIMENTAL STUDY

A. Setup

As shown in Fig. 7, the experimental setup consists of a keyboard, a speaker, a human driver and the modified TORCS game containing our trained LSTM network and a voice interface. The human subject drives the ego-vehicle, i.e., a racing car by using the keyboard arrows. Each arrow is assigned to positive acceleration, negative acceleration, and steering. The monitor in Fig. 7 displays the current driving situation to the driver, and our driving assistant companion informs the driver through a speaker what driving should be done in the current driving situation. The human driver then drives the car in the modified TORCS game with the help of the voice assistance to minimize the lap time. All experiments were performed on a workstation equipped with 16GB RAM, Intel Core i5-7600 CPU@3.5GHz running 64-bit Ubuntu 16.04, and GeForce GTX 1050.

Fig. 7. The experimental setup that consists of a modified TORCS game, a keyboard controller, a speaker and a human driver.

B. Procedure

A total of 16 human subjects between the ages of 21 and 27 participated in the experimental study. The subjects are randomly divided into two groups. The group without the voice assistance consist of 6 males and 2 females. This group is a control group and is called Group 1. The other group with the voice assistance consists of 6 males and 2 females. This group is called Group 2. All participants are newcomers to the TORCS game. All subjects drive 12 laps in the game. In what follows, we describe the experimental procedure.

- Intro: In the first three laps, both groups play the TORCS game without the voice assistance just to be familiar with the game.
- Train: During the next 7 laps, Group 2 is provided with the voice assistance to see the efficacy of our assistant companion as compared to Group 1. This period is called the training section.
- Test: The last two laps, both groups play without the voice assistance to gauge the performances after the training section.

The driving performance is rated by the total lap-time. The lap time can be increased if there is a crash during driving or if the driver follows a non-optimal path [22].

C. Statistical Analysis Method

We use the Mann-Whitney U-test to determine the correlation between the results of the two groups. This is a statistical test that compares two independent groups when the sample size of the group is less than 30 and the normality is not met [27]. The p-value used in this test will determine whether the difference between the two groups is significant.

V. RESULTS

A. Classification Results

In this section, we calculate the accuracy, the precision and the recall for our multi-class classification problem solved in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2861739, IEEE Transactions on Industrial Informatics

TABLE II
CONFUSION MATRIX FOR LSTM NETWORK CLASSIFIER

True class	Predicted class					
	SL	TR	SR	TL	AC	BC
SL	57482	44	322	1	551	0
TR	482	39146	9	225	38	100
SR	1146	38	25452	389	975	0
TL	54	132	913	19634	3	64
AC	363	1	57	1	33178	0
BC	0	5	0	269	0	22526

TABLE III
PRECISION AND RECALL

Class	Precision (%)	Recall (%)
SL	96.6	98.4
TR	99.4	97.9
SR	95.1	90.9
TL	95.7	94.4
AC	95.5	98.7
BC	99.3	98.8

Section III. The accuracy is the correct classification rate. The accuracy is defined as

$$\text{Accuracy} := \frac{\sum TP_i}{Total\ \#\ of\ data} \times 100(\%), \tag{14}$$

where TP denotes the true positive and i denotes the index of the class.

The precision is an agreement of the data class label with that of classification [28]. The precision is defined as

$$\text{Precision}_j := \frac{TP_j}{FP_i + TP_j} \times 100(\%), \tag{15}$$

where FP denotes the false positive and j denotes the column index of the confusion matrix.

The recall is the effectiveness of the classifier to identify the class label [28]. The recall is defined as

$$\text{Recall}_k := \frac{TP_k}{FN_k + TP_k} \times 100(\%), \tag{16}$$

where FN denotes the false negative and k denotes the row index of the confusion matrix. Table II shows the confusion matrix obtained by the classification results using the LSTM network on the test data set. The total number of the classification tests is $203,600$. Table III demonstrates the good performance of the proposed LSTM network in terms of the precision and recall values for all classes.

B. Experimental results

To investigate whether there is an initial correlation between lap times from two groups, we averaged individual lap times from lap 1 to lap 3 and then calculate the p-value via Mann-Whitney U-test. The p-value (two tail) is 0.8336 (> 0.05) and does not establish an initial correlation between lap times from two groups.

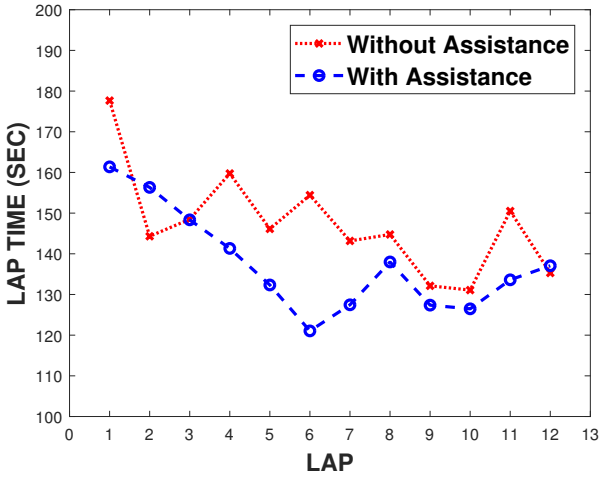

Fig. 8. Average lap times vs laps of each group

1) Hypothesis I (Performance enhancement): The first hypothesis we consider is that the group (Group 2) with the voice assistance performs better than the control group (Group 1). To test the hypothesis, we average the lap times for each lap over subjects per each group as shown in Fig. 8. Note that from lap 4 to lap 10, i.e., the training section, the driving assistant companion system applies to Group 2. Figure 8 clearly shows that Group 2 performs better than Group 1 at the training section. The averaged lap time in the training section is 147.30 sec for Group 1 and 130.95 sec for Group 2. The p-value (two tailed) through Mann-Whitney U-test in the training section is 0.0181 (< 0.05). Therefore, the difference of the average lap times between two groups in the training section is statistically significant. The first hypothesis is supported by the experimental evidence.

2) Hypothesis II (Reliability): The second hypothesis we consider is that the performance of the group with voice assistance (Group 2) is more reliable than that of the control group (Group 1). We show the standard deviation error bars between the subjects' lap times for each lap to see the variability of the lap times of the subjects. When the variability is low, the performance reliability is high. The standard deviation is computed by the following formula.

$$S := \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} |A_i - \mu|^2}, \tag{17}$$

where S is the standard deviation, N is the number of subjects in each lap, A_i is the lap time of each subject, and μ is the average lap time. Since there is a difference in the help of assistive companion during the training section, it is necessary to review the variability during that period to gauge reliability. Figures 9 and 10 show that lap times of Group 2 are more reliable than those of Group 1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2861739, IEEE Transactions on Industrial Informatics

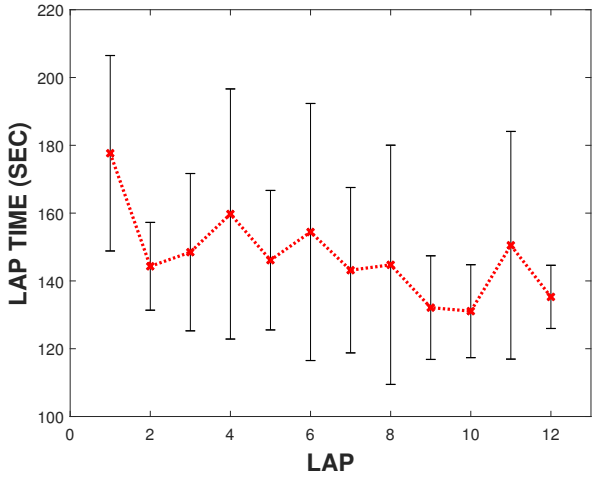

Fig. 9. Means and standard deviations ($\pm\sigma$) of Group 1

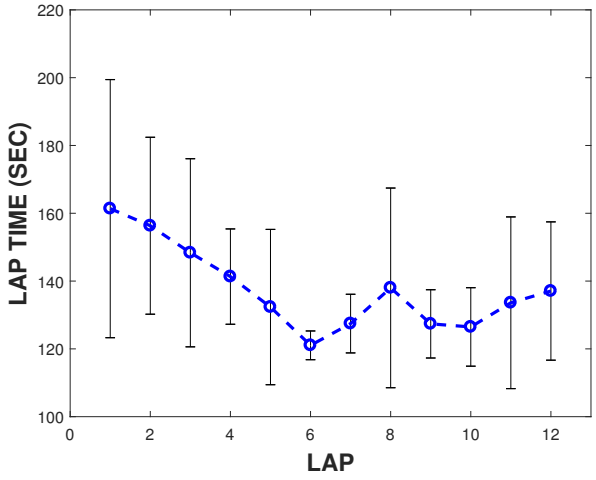

Fig. 10. Means and standard deviations ($\pm\sigma$) of Group 2

We consider the standard deviations as new random variables for the U-test. The means of standard deviations at the training section are 28.11 and 15.418 in Groups 1 and 2, respectively. We analyze the difference of the reliability between two groups by using U-test with standard deviations. The p-value for the data in the training section is 0.0350 (< 0.05). Therefore, the difference of the standard deviations in the training section between the two groups is statistically significant. The second hypothesis is also supported by the experimental evidence.

3) Learning Curve: Figure 11 illustrates the fitted learning curve [29] for Group 1, Group 2, and all subjects. All curve fittings were performed with the minimum root mean square (RMS) error using the power law. The R-square values are 0.72, 0.58, and 0.83 for Group 1, Group 2, and all subjects, respectively.

The learning rate [29], [30] is defined as

$$\text{Learning rate} := \frac{T_a}{T_b} \times 100(\%), \tag{18}$$

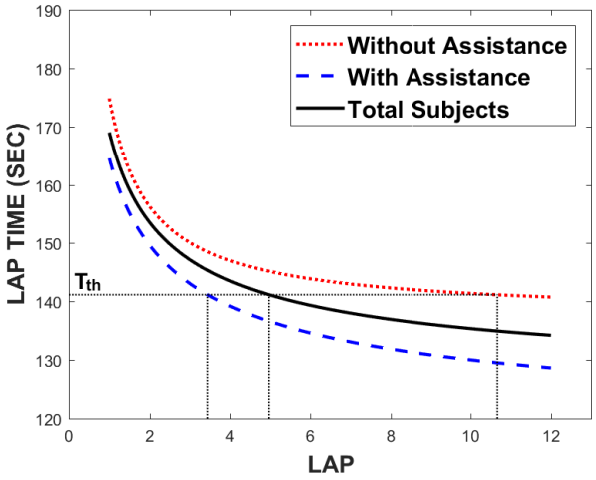

Fig. 11. Learning curves of the total subjects and each group

where T_a is the lap time after the training, and T_b is the lap time before the training. When considering only the training section, the learning rate of Group 1 is 3.8% and that of Group 2 is 6.6%. Overall, from lap 1 to lap 12, the learning rate of Group 1 is 19.49%. The learning rate of Group 2 is 21.96% which is larger than that of Group 1. Therefore, we have a consistent tendency of having higher learning rates from Group 2.

The learning speeds [29], [30] can be defined as follows.

$$\text{Learning speed} := \frac{T_{th}}{L_{th}}, \tag{19}$$

where T_{th} is the threshold lap time, and L_{th} is the number of laps to reach the threshold lap time. The threshold lap time can be determined from the learning curve of all subjects [30]. The learning rate of the total learning curve is 20.59%, and its 80% is 16.47% [30]. The lap time at 80% of the total learning rate is determined as the threshold lap time. From our data, T_{th} is 141.16 seconds. We assume the lap as a continuous value and obtain the lap of each group to reach the threshold lap time. Group 2 takes 3.44 laps (L_{th}) to reach T_{th}, and Group 1 takes 10.66 laps (L_{th}). Hence, Group 2 shows a faster learning speed of 41.03 as compared to that of Group 1 (13.24).

VI. DISCUSSION

Our driving assistant companion uses the LSTM network with online stream of range finding sensor measurements in order to classify situational events. It shows the better accuracy (96.96%) than that of IB1 algorithm with global positioning system (GPS) based mobile sensors [31] (89.85%). Mu noz-Organero *et al.* [12] provides the driver with the voice assistance on how to release accelerator pedal to reduce fuel consumption. Mu noz-Organero's work is similar to ours in a sense of the voice assistance. However, they use a traffic signal detecting algorithm with the stream of the vision data while we use the LSTM network with the stream of the range finder measurements.

In addition, we tried classification with the same dataset using an SVM using linear kernels, one of the traditional

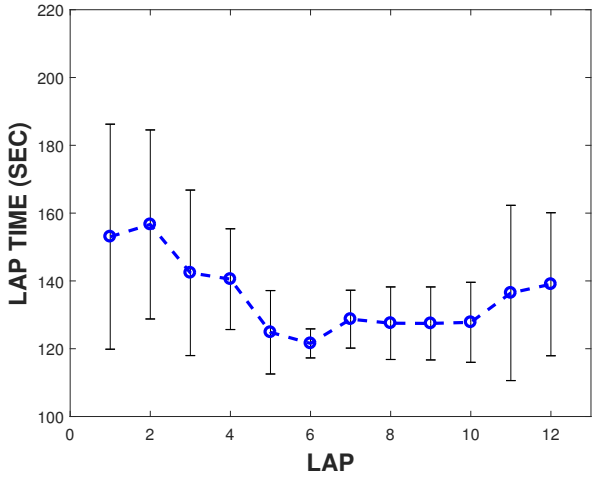

Fig. 12. Means and standard deviations ($\pm\sigma$) of Group 2 without an outlier

classification methods in machine learning. The accuracy of the SVM model was 83% while that of our LSTM network was 98%, which shows the better performance. The LSTM network shows the better results because it has ability to process and predict time-series sequences without forgetting unimportant information. The LSTM network achieves state-of-the-art results in time-series problems [23].

The total number of collected data for classification learning is 12,710. This is sufficient number for training our model that is validated by using 10-fold cross-validation with the average accuracy of 98.27% as shown in Section 3-B.

The experimental results present that the performance and reliability of Group 2 outperform those of the control group. Looking at Fig. 10, the standard deviations are small at laps 6, 7, 9, and 10, but unusually large at lap 8 due to an outlier. An outlier with an abnormally slow lap time causes high average lap time and the large standard deviation. Figure 12 illustrates standard deviations of Group 2 without the outlier. In the training section, the standard deviation is decreased after the outlier is removed.

At the test period, i.e., laps 11 and 12 in Fig. 12, the standard deviations and the average lap times increase again. We suspect that the decreased performance and reliability might be due to the evaluation anxiety [32]. Subjects with high evaluation anxiety may show poor performance, resulting in slower average lap times and bigger standard deviations. Indeed, most of the subjects expressed that they felt a burden of trying to be the best at the last two laps as the final test.

In Fig. 11, the difference in the learning speed might be due to differences in applied learning factors [33]. The difference may be due to the fact that two factors, multiple practice trials and the voice assistance, are applied to Group 2 while only the multiple practice trials are given to Group 1.

VII. CONCLUSIONS

In this paper, the LSTM network is used to compose a driving assistance companion system using a voice interface. It provides drivers with useful information to enhance learnability, driving performance, and reliability. The range finding sensor is used for real-time prediction without map information. The experimental study is designed using the TORCS to demonstrate the efficacy of the proposed assistant companion.

Our assistant companion technology can be applied to various problems in industry such as training programs for automobiles, motorized vehicles, remotely controlled vehicles, drones, and surgical robots. The future work is to apply our assistant companion on a real car by using the LSTM network with the real-time measurements from LiDAR sensors.

REFERENCES

[1] V. HERNDON, "Retrieved august 30, 2017, from https://www.audiusa.com/newsroom/news/press-releases/2017/01/audi-and-nvidia-to-bring-fully-automated-driving-in-2020," January 05, 2017.

[2] D. Etherington, "hyundai looks to increase collaboration with google. techcrunch, techcrunch, 17 aug. 2016, techcrunch.com/2016/08/17/hyundai-looks-to-increase-collaboration-with-google/." 30 Aug. 2017.

[3] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[4] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2004, pp. 70–75.

[5] M. Yamamoto, A.-T. Hoang, and T. Koide, "Speed traffic sign recognition algorithm for real-time driving assistant system," in *Proceeding of the 18th Workshop on Synthesis and System Integration of Mixed Information Technologies (SASIMI 2013)*, 2013.

[6] K. A. Brookhuis, D. De Waard, and W. H. Janssen, "Behavioural impacts of advanced driver assistance systems–an overview," *European Journal of Transport and Infrastructure Research*, vol. 1, no. 3, pp. 245–253, 2001.

[7] F. Biral, M. Da Lio, and E. Bertolazzi, "Combining safety margins and user preferences into a driving criterion for optimal control-based computation of reference maneuvers for an adas of the next generation," in *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE.* IEEE, 2005, pp. 36–41.

[8] B. G. Simons-Morton, F. Guo, S. G. Klauer, J. P. Ehsani, and A. K. Pradhan, "Keep your eyes on the road: Young driver crash risk increases according to duration of distraction," *Journal of Adolescent Health*, vol. 54, no. 5, pp. S61–S67, 2014.

[9] I. E. González, J. O. Wobbrock, D. H. Chau, A. Faulring, and B. A. Myers, "Eyes on the road, hands on the wheel: thumb-based interaction techniques for input on steering wheels," in *Proceedings of Graphics Interface 2007*. ACM, 2007, pp. 95–102.

[10] B. Rosen, "Amazon's alexa to hit the road: a safer or more dangerous driving experience? retrieved august 28, 2017, from https://www.csmonitor.com/technology/2017/0105/amazon-s-alexa-to-hit-the-road-a-safer-or-more-dangerous-driving-experience," 2017.

[11] C. Thompson, "I used amazon's alexa to control a ford fusion and i'm convinced it's the future of driving retrieved august 29, 2017, from http://www.businessinsider.com/amazon-alexa-future-driving-2017-2," 2017, March 21.

[12] M. Muñoz-Organero and V. C. Magaña, "Validating the impact on reducing fuel consumption by using an ecodriving assistant based on traffic sign detection and optimal deceleration patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 1023–1028, 2013.

[13] K. J. Anstey, J. Wood, S. Lord, and J. G. Walker, "Cognitive, sensory and physical factors enabling driving safety in older adults," *Clinical psychology review*, vol. 25, no. 1, pp. 45–65, 2005.

[14] B. Lee, O. Kwon, I. Lee, and J. Kim, "Companionship with smart home devices: The impact of social connectedness and interaction types on perceived social support and companionship in smart homes," *Computers in Human Behavior*, vol. 75, pp. 922–934, 2017.

[15] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*. Presses universitaires de Louvain, 2015, p. 89.

[16] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*. IEEE, 2015, pp. 1–7.

[17] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2016, pp. 130–139.

[18] T. Uusitalo and S. J. Johansson, "A reactive mutli-agent approach to car driving using artificial potential fields," in *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*. IEEE, 2011, pp. 203–210.

[19] J. Muñoz, G. Gutierrez, and A. Sanchis, "A human-like torcs controller for the simulated car racing championship," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 473–480.

[20] S. Tognetti, M. Garbarino, A. T. Bonanno, M. Matteucci, and A. Bonarini, "Enjoyment recognition from physiological data in a car racing game," in *Proceedings of the 3rd international workshop on Affective interaction in natural environments*. ACM, 2010, pp. 3–8.

[21] E. J. Ogden and H. Moskowitz, "Effects of alcohol and other drugs on driver performance," *Traffic injury prevention*, vol. 5, no. 3, pp. 185–198, 2004.

[22] M. Mühlmeier and N. Müller, "Optimisation of the driving line on a race track," *AutoTechnology*, vol. 3, no. 2, pp. 68–71, 2003.

[23] M. A. Zaytar and C. El Amrani, "Sequence to sequence weather forecasting with long short-term memory recurrent neural networks," *International Journal of Computer Applications*, vol. 143, no. 11, 2016.

[24] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 130–139.

[25] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao *et al.*, "Relation classification via convolutional deep neural network." in *COLING*, 2014, pp. 2335–2344.

[26] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4, pp. 185–196, 1993.

[27] N. Nachar *et al.*, "The mann-whitney u: A test for assessing whether two independent samples come from the same distribution," *Tutorials in Quantitative Methods for Psychology*, vol. 4, no. 1, pp. 13–20, 2008.

[28] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[29] L. Diawati *et al.*, "The impact of learning on assembly line capacity (a case study of an indonesia car factory)," in *Management of Innovation and Technology (ICMIT), 2012 IEEE International Conference on*. IEEE, 2012, pp. 320–324.

[30] N. E. Raine and L. Chittka, "The correlation of learning speed and natural foraging success in bumble-bees," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 275, no. 1636, pp. 803–808, 2008.

[31] P. Tchankue, J. Wesson, and D. Vogts, "Using machine learning to predict the driving context whilst driving," in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*. ACM, 2013, pp. 47–55.

[32] S. I. Donaldson, L. E. Gooler, and M. Scriven, "Strategies for managing evaluation anxiety: Toward a psychology of program evaluation," *American journal of evaluation*, vol. 23, no. 3, pp. 261–273, 2002.

[33] B. J. Zimmerman and M. Martinez-Pons, "Construct validation of a strategy model of student self-regulated learning." *Journal of educational psychology*, vol. 80, no. 3, p. 284, 1988.

Jehyun Park received his B.S. degree in Mechanical Engineering from Yonsei University at Seoul, South Korea in 2017. He is working toward a Ph.D degree in Machine Learning and Control Systems Lab at Yonsei University. His research interests include machine learning, robotics, parameter estimation, intelligent systems and control with applications on an artificial intelligence-based service robot and deep learning-based autonomous driving control systems utilizing situation awareness and its uncertainty.

Hojoon Son received his B.S. degree with double majors in Mechanical Engineering and Electrical & Electronic Engineering from Yonsei University at Seoul, South Korea in 2017. He is currently a member of Knowledge Based Design Lab at Yonsei University as an MS student. His research interests include machine learning, data science, intelligent design, robotics, control, and artificial intelligence systems.

Jisuk Lee received his M.S. degree in Aerospace Engineering from Korea Institute of Science and Technology (KAIST) at Daejeon, South Korea in 2009. He is a Ph.D student in Machine Learning and Control Systems Lab at Yonsei University and CTO of Motion Device Inc. which is a VR theme park company. His current research interests include leveraging new technologies; VR and AI for improving existing VR entertainment simulators and offering new solutions in the VR theme park area.

Jongeun Choi received his Ph.D. and M.S. degree in Mechanical Engineering from UC Berkeley in 2006 and 2002, respectively. He also received a B.S. degree in Mechanical Design and Production Engineering from Yonsei University at Seoul, South Korea in 1998. He currently heads up Machine Learning and Control Systems Lab and is an Associate Professor with the School of Mechanical Engineering at Yonsei University at Seoul, South Korea. Prior to joining Yonsei University, he worked for 10 years as an Associate Professor and an Assistant Professor with the Departments of Mechanical Engineering and Electrical and Computer Engineering at the Michigan State University during 2012-2016 and 2006-2012, respectively. His current research interests include machine learning, systems and control, system identification, and Bayesian methods, with applications to autonomous robots, self-driving vehicles, mobile sensor networks, (physical) human and robot interaction, and biomedical problems. He is an Associate Editor for IEEE Robotics and Automation Letters (RA-L), Journal of Dynamic Systems, Measurement and Control (JDSMC), and International Journal of Precision Engineering and Manufacturing (IJPEM). He co-organized and co-edited special issues on Stochastic Models, Control, and Algorithms in Robotics in JDSMC, during 2014-2015, and Deep-Learning Based Sensing Technologies for Autonomous Vehicles in Sensors in 2018. He received the Best Conference Paper Award at the 12-th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) 2015. His papers were finalists for the Best Student Paper Award at the 24th American Control Conference (ACC) 2005 and the Dynamic System and Control Conference (DSCC) 2011 and 2012. He is a recipient of an NSF CAREER Award in 2009. Dr. Choi is a member of ASME and IEEE.