Amirkabir University of Technology
(Tehran Polytechnic)
Vol. 47 - No. 2 - Fall 2015, pp. 21- 29

Amirkabir International Journal of Science& Research
(Modeling, Identification, Simulation & Control)
(AIJ-MISC)

Neuro-ACT Cognitive Architecture Applications in Modeling Driver's Steering Behavior in Turns

A. Dehban[1], M. B. Menhaj[2,3]*, and A. Sajedin[4]

1- MScStudent, Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran
2- Professor, Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran.
3- Professor, QIAU's Incubator Center of Technology Units, Qazvin, Iran
4- PhD Student, Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

ABSTRACT

Cognitive Architectures (CAs) are the core of artificial cognitive systems. A CA is supposed to specify the human brain at a level of abstraction suitable for explaining how it achieves the functions of the mind. Over the years a number of distinct CAs have been proposed by different authors and their limitations and potentials were investigated. These CAs are usually classified as symbolic and sub-symbolic architectures. In this work, a novel hybrid architecture is proposed that encompasses a symbolic part (i.e. ACT-R) to explain the controlled aspects of behavior and a sub-symbolic part (i.e. Artificial Neural Networks) to describe automated skills. In order to demonstrate the capabilities of the proposed model, an experiment was conducted in which, a rather complex real life task was carried out by the model and its result were compared with those of human participants. Simulation results have shown promising capabilities of the new architecture in modeling complex human behavior.

KEWORDS

Cognitive Architecture, Neural Networks, Cognitive Dynamic Systems, Driver Modeling, Neuro-ACT

*Corresponding Author, Email: menhaj@aut.ac.ir

1. INTRODUCTION

In recent years computational-based modeling has emerged as a powerful tool for studying the complex task of driving which allowing many researchers to simulate driver behavior [1-4]. During the past decade, lots of efforts have been made to provide a comprehensive model of driver behavior [5-8]. Driving is a very common everyday task and yet a very complex as well as hazardous one [9-13]. Modeling driver behavior in a cognitive architecture can shed some light on the embodied procedures taken by drivers. On the other hand, due to big demands in cognitive abilities, driving is a good situation in which the capabilities and limitations of cognitive architectures can be tested [14-16]. Another benefit of having an integrated model capable of reproducing actions of drivers is utilizing this model in place of the human subjects in order to facilitate an evaluation process in designing advanced driver assistance systems [17-19].

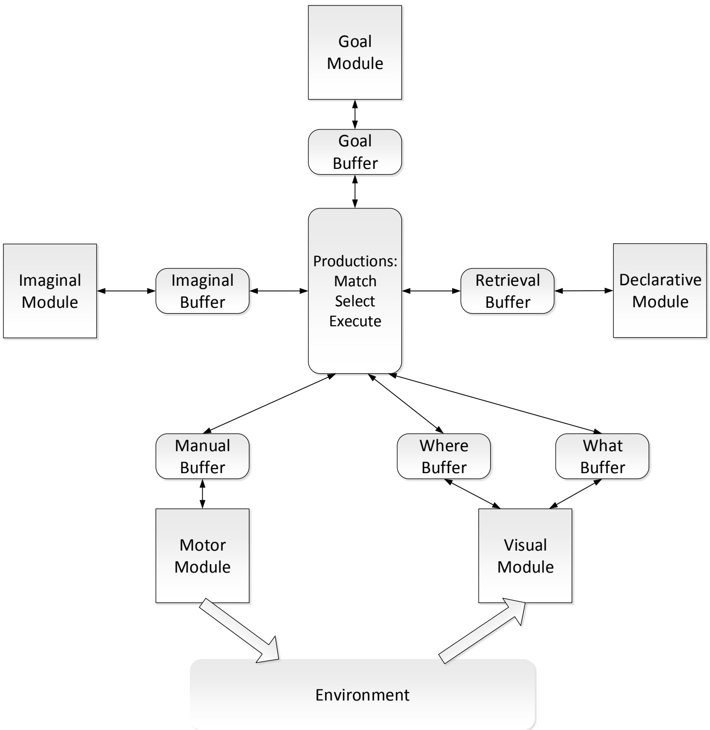

Fig. 1. Some of ACT-R buffers and modules

Such systems can enhance their capabilities if they could understand and characterize different aspects of driver's behavior [20-23].A cognitive model which tries to explain driving must be able to answer the questions like how decision making or situation awareness is achieved,what are the importantcognitive bottlenecks

and limitations are, how different inputs from the environment are perceived by drivers and how he/she can use his/her motor skills to manipulate the vehicle, and etc. [24-27].

A strong psychological theory is necessary for providing proper answers to the aforementioned issues. Computational models of human cognition due to 1) their high level of clarity and completeness, and 2) their better explanation and evaluations, can provide a reliable framework for modeling the different aspects of human cognition [19].

It should be noted that it is not all things that happen while driving can be explained solely through a cognitive architecture [20]. Much of driving is composed of lower level automated skills and reflexive behaviors which are out of the scope of cognitive architectures. Distinguishing these two processes is a common practice and cognitive architectures are usually flexible enough to allow execution of such tasks. It must be noted that any process of sufficient complexity involves a complex interplay between both controlled (conscious) and automatic processes[21].

Salvucci [2] has proposed an integrated driver model using Adaptive Control of Thought – Rational (ACT-R) cognitive architecture and uses a PI like controller to account for those reflexive behaviors. Although the PI controller can produce human like behavior [12] it cannot be stated as realistic and it lacks a biological or psychological explanation. Furthermore, his claim for modeling a skilled driver (100 references of memory chunks for easy retrieval) suffers from an unrealistic explanation.

Mihalyi [20] used a fuzzy logic compensator instead of Salvucci's PI controller. Although his model can explain the reason for this controller to work, in our opinion fuzzy logic cannot be accounted for automated skills. Fuzzy logic is developed to model expert's knowledge and provides a means to compute with words, but it never claims that what really happen in the brain are fuzzy computations.

In this paper, a computational driver model developed in the ACT-R (Adaptive Control of Thought-Rational) cognitive architecture and by taking advantage of an Artificial Neural Network (ANN)as the basis of the underlying control mechanism, focused on the component processes of control, and decision making in a multilane highway environment. MLPs are a fully-connected feedforward artificial neural network which learns a mapping between a set of input samples and their corresponding target classes. The MLPs is in fact an extension of the Perceptron neural network. Each node in a MLPs neural network represents a neuron which is usually considered as a nonlinear processing element.

We have used an (Multi-Layer perceptron) MLP for modeling driver's automated skills and predict driver's behavior, based on Salvocci's simulation environment to acquire training data for the MLP. In our experiments, the drivers could see the environment via monitor screen and also could control the steering wheel of the vehicle using a cell phone device. Adding on, an application was developed to send the pitch angle of the phone in determined time steps. The participants are four graduate students aged between 23-32, which have trained to drive in our environment and used recorded data to train the Multilayer perceptron (MLP).

The reminder of this paper is organized as follows: In Section 2, we describe the relevant ACT-R components which are effectively used in the proposed model for the integrated driver behavior. Section 3 is devoted to introducing ANNs and explain how they can potentially perform better than fuzzy logic controllers in modeling reflexive behavior, followed by Section 4, in which a realization of an ANN as well as integration of this controller in ACT-R are fully presented. In order to validate the proposed model we compare and discuss its behavior with that of human drivers in Section 5. Finally, Section 6 concludes the paper.

2. ACT-R COGNITIVEARCHITECTURE

A cognitive architecture is a framework for specifying computational behavioral models of human cognitive performance. The architecture consists of the abilities and constraints of the human system such as memory and recall, learning, perception, and motor action [28-30]. A cognitive architecture can psychologically confirm the cognitive models which are developed in the framework [2], [14]. ACT-R (Adaptive Control of Thought-Rational), is a hybrid architecture based on chunks of declarative knowledge and condition-action production rules that operate on these chunks. ACT-R is a computational model of human brain that describes how cognitive functions are realized in brain and it can also be used to replicate human behavior. ACT-R is mainly used to explain and reproduce the experimental data that is obtained in psychology experiments [12], [14]. ACT-R contains modules interacting through their buffers that are organized by a central controller (see Fig. 1).

The ACT-R is simultaneously a rigorous theory of human cognition and a working framework in which to build computational models of human behavior which posits two separate but interacting knowledge stores. The first type of knowledge, declarative knowledge, is made up of chunks, or small logical units, of symbolic information. Declarative chunks can encode simple facts, current goals, and even ephemeral situational. Chunks are associated with sub-symbolic parameters that encode continuous valued properties of each chunk [12], [14].

The second type of knowledge is procedural knowledge which is made up of rules representing procedural skills that manipulate declarative knowledge as well as the environment. When all conditions match and the rule fires, rule actions can add to or alter declarative memory, set a new current goal, and/or issue perceptual or motor commands. Also, ACT-R has the ability to perform some processes in parallel such that [2], [12] and [14]. At the same time, ACT-R places certain limitations and constraints on models that mimic the constraints of the human system. One of the important constraints for the driver model is that although perceptual and motor processes can run in parallel with cognition, the cognitive processor itself is serial and, can think just one thing at a time. The cognitive processor is responsible for collecting all information from perceptual modules and issuing all motor commands, and thus it serves as the central bottleneck for behavior. This fact is critical for applications such as predicting driver distraction.

A. Goal Module

The goal module is responsible for keeping track of one's internal intentions and goals. This module can control the flow of thought in order to achieve a goal-directed behavior. Consider a task of getting to work in your car. This task can be decomposed into several abstract steps like dressing, getting to your car, starting the engine and driving to your workplace, parking your car near destination, etc. The ACT-R cognitive architecture deploys the goal module 1) to be aware of its current task and 2) to know which tasks should be considered next in order to achieve its overall goal (see Fig. 1).

B. Imaginal Module

Imaginal module is used to hold the mental representation of the problem. When new information about the current task in different buffers exists (e.g. visual module buffers) this information is organized in the imaginal buffer. This organized information will later be saved in the declarative module for the successive retrievals.

The process of information organizing in imaginal module is indeed time consuming. When there is a request to create a new chunk in the imaginal buffer or to modify its current contents, some amount of time should pass before these requests are fulfilled. In the ACT-R 6.0 implementation this time can be a fixed constant or a random variable.

C. Declarative Module

Declarative module is the ACT-R's implementation of declarative memory. Declarative memories are the kind of memories that can be declared, such as the name of one's fifth grade teacher or the name of the South African anti-apartheid leader. One can view declarative module like a large warehouse containing a vast amount of information. However, since humans may need some efforts or struggling to retrieve past information from their memory, retrieving information from declarative module must also be able to reflect these problems (see Fig. 1).

In ACT-R literature pieces of information are called *chunks*. Each chunk that is stored in the declarative module is associated with an activation value. This value is calculated according to (1). This equation is used in the proposed model of driver to retrieve memories from his/her past.

$$A_i = B_i + \varepsilon \tag{1}$$

In the above equation, ε reflects the noise value in retrieving a memory and B_i is the *Base Level Activation* (BLA) of chunk *i*. This value depends on how many times this chunk has been practiced according to the following equation:

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \tag{2}$$

wheret_j is the time since the jth practice of chunk i.d is the decay rate of the chunk which reflects how amemory is forgotten and almost always is set to a default value of 0.5. In order to simplify the calculation of BLA, one can use an alternative formula instead of .

$$B_i = \ln\left(\frac{n}{1-d}\right) - d \times \ln\left(L_i\right) \tag{3}$$

L_i is the time passed since creation of chunk i. Equation3, is the default formula for calculation of BLA in the ACT-R 6.0 implementation and it can be set by an optimized learning parameter. ε in[2]contains two sources of noise. One is a permanentnoise which is associated with the chunk at the time of creation and the other is an instantaneousnoise that is calculated on each retrieval attempt. Both of these noises are generated according to a zero-mean logistic Probability Distribution Function (PDF). The variance σ^2 of this PDF is related to noise parameter s according to the following equation:

$$\sigma^2 = \frac{\pi^2}{3} s^2 \tag{4}$$

Hence the instantaneous noise parameter s and permanent noise parameter s determine the variance of instantaneous and permanent noise, respectively.The activation of each chunk determines both probability and latency of its retrieval defined respectively in (5) and (6).

$$P_i = \frac{1}{1 + e^{-(A_i - \tau)/s}} \tag{5}$$

In the above, P_i is the probability of retrieval of chunk, *i* is a constant called retrieval threshold and reflects the attempt one makes in order to retrieve a memory and *s* is the instantaneous noise parameter

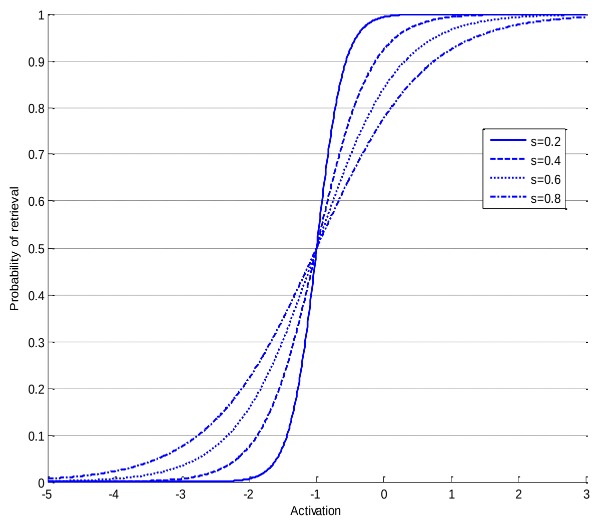

Fig. 2. .Probability of retrieval for different values of noise parameter

If a chunk is selected for retrieval, the time it takes to be retrieved is also related to its activation:

$$T_i = Fe^{-A_i} \tag{6}$$

where, T_i is the time it takes for chunk *i* to be retrieved and *F* is a constant called *latency factor*. Anderson et al. [8] proposed the following rule of thumb relation between the *latency factor* and the *retrieval threshold*:

$$F \approx 0.35e^{\tau} \tag{7}$$

D. Motor Module

ACT-R motor module models the basic timing behavior of human motor system when dealing with computer interface using mouse and keyboard [2], [12]. Note that the architecture is flexible enough to allow for arbitrary actions when dealing with real world interfaces (see Fig. 1).

E. Visual Module

This module is used to view objects in the visual field. As many researchers have shown, two buffers are associated with the visual module. One is responsible for encoding the position of objects in the visual field and the other one to encode the nature of it. It is assumed that the visual module performs a parallel process in order to encode the position and some basic properties of all objects visible but only certain objects that had been subject to attention are known to the system. Request to the *where module* includes a few attribute-value pairs (e.g. vertical down) resulting in a chunk that represents the exact position and some basic properties of that object. If further information about the object is needed, this chunk is used to make a request to the *what buffer*. This request shifts attention of the visual module to specified location and additional information about that object is retrieved. Due to the parallel processes in the visual module to encode the location of visible objects, these requests are fulfilled instantly though the time it takes to shift visual attention is related to the distance between the current point of the attention and the new request.

F. Productions

Productions constitute the procedural memory of ACT-R cognitive architecture. They are *if-then* rules that respond to patterns presented in different buffers. The *then* part of each rule consists of a few requests sent to various buffers so that the corresponding modules update those buffers in response to these requests. According to the ACT-R theory, only one rule can fire at a time and there must be a 50*ms* delay before firing of the subsequent rules

3. Artificial Neural Networks For Automated Skills

ANNs are one of the earliest attempts to provide a computational model of human brain. Multilayer perceptron (MLP) networks are largely used to approximate unknown functions and relations among data. In order to answer the questions of how to model human reflexive behavior, two approaches seemed feasible: ANNs and fuzzy logic controllers[20]. In this section we describe the advantage of using ANNs over fuzzy logic computations for modeling driver's automated skills.

MLPs is a fully-connected artificial neural network which learns a mapping between a set of input samples and their corresponding target classes. The MLPs is in fact an extension of the Perceptron neural network. Each node in a MLPs neural network represents a neuron which is usually considered as a nonlinear processing element. We have used an (Multi-Layer perceptron) MLP for modeling driver's automated skills and predict driver's behavior, based on Salvocci's simulation environment to acquire training data for the MLP.

A. Advantages Of Using ANNs

ANN's advantage becomes obvious when one takes a look at how automaticity develops. According to [14] there are three stages in the acquisition of skills: cognitive stage, associative stage and automatic stage. The cognitive stage is seen as tightly linked to verbal descriptions. For instance, driving instructors usually provide some rules when explaining to a learner when to change gears. The second stage (associative stage) is characterized by reduction of verbal mediation.

The final stage is when automatic and verbalization is no longer needed or possible. It must be pointed out that the automaticity is not assumed to result exclusively from a process of skill acquisition.

Linguistic variables are vastly used in initial cognitive stage but unless a driver has reached the final stage, he/she is not allowed to drive without assistance of a supervisor. Therefore computation with words does not seem plausible when modeling behavior of a skilled driver. On the other hand, automated behavior is assumed to form a particular circuitry in human brain and ANNs are a simple computational tool to mimic the behavior of that brain circuit.

There is also another reason for the proposed ANN to work as stated below. Michon [24] identified three classes of task processes for driving: operational processes that involve manipulating control inputs for stable driving, tactical processes that govern safe interactions with the environment and other vehicles, and strategic processes for higher level reasoning and planning. These operational processes or as we call it, *automated skills*, can be viewed as direct mapping from environment stimuli to control outputs. Generally speaking, if proper data is available, ANNs are superior over fuzzy logic computations in approximating unknown mappings, indisputably.

B. Realizing An ANN To Reproduce Human Steering Behavior

In order to model automated driving skills, Salvucci and Gray in ref [22] suggested that human drivers use information of two salient points to guide their steering, namely near point and far point.

In this paper we have used the time between control updates Δt and also longitudinal speed of the vehicle v_x when facing a turn in the road, as the additional information in which our model to be able to performs accurate driver behavior prediction.

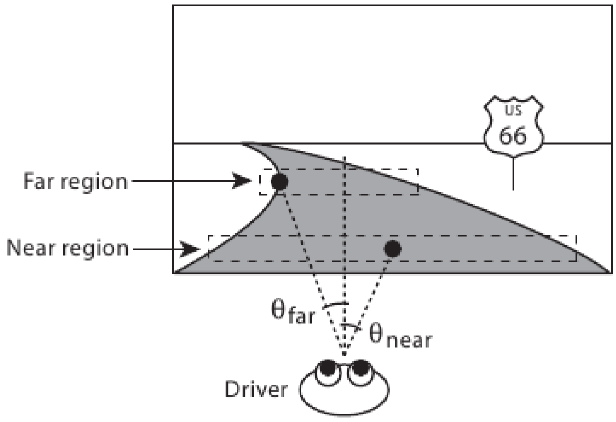

Fig. 3. Near point and far point when facing a turn in the road

It must be pointed out that the curvature information of the turn is not explicitly included in the training data of ANN, since this information can be calculated using $\Delta\theta_{far}, \Delta t, v_x$.

Furthermore, drivers usually do not know in advance about the exact properties of road curves fig.4. illustrates the inputs, structure and output of the proposed ANN.In fig.4 $\Delta\delta$ is the change in steering angle to compensate for the curve of the road.

C. Putting It All Together

Now that all the components of the system are described, it is the time to put them together to model the driver's steering behavior.

First, we describe the procedural memory of ACT-R cognitive architecture. Next, a brief description of the data acquisition system in a simulated environment is presented which are later have been used to train the ANN described in the previous section. The model can then drive in the same driving simulator.

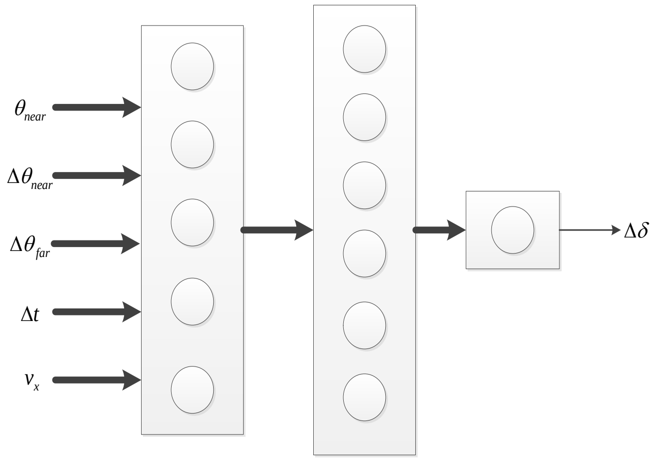

Fig. 4. Structure, inputs and output of the proposed MLP

D. Procedural Memory To Pass A Curve In The Road

As discussed earlier, procedural memory of ACT-R is composed of condition-action pairs where conditions are patterns in different buffers and actions are change requests to those or other buffers. Fig. 5 depicts the flowchart of rules controlling behavior of the model.

Table 1 is the procedural memory of the driver model and explains how different buffers of Fig. 1 are utilized to achieve the goal directed behavior. Another module, called Eval module, is implemented in ACT-R 6.0 to allow for execution of the user defined functions.

This module is used to calculate the steering angle change $\Delta\delta$ via the proposed ANN. The last rule of table1 fires when no memory of past could be retrieved. This will happen in two situations, when the model is run for the first time or when the model does not try hard enough to remember its past for example a higher retrieval threshold is used.

4. Estimation Of ACT-R Parameters

Tuning various parameters of cognitive architectures for fitting to the collected data is usually discouraged and good cognitive models should have as few free parameters as possible [18]. In order to comply with this general rule of cognitive modeling, almost all parameters of our model are the default ones that there is a consensus in their values in the ACT-R community. The only parameter that we have changed to reflect the abilities of a skilled driver is the Imaginal delay parameter which due to low information volume that is placed in the imaginal buffer, we lowered this value to 35ms. In addition, as described before, ACT-R is not integrated with a default mechanism of hand movement for steering control, thus we assumed it takes 100ms for hands to modify the steering angle of vehicle. These parameters were fixed before we gathered our data.

5. SIMULATION ENVIRONMENT AND DATA GATHERING

Salvucci has made available the source code of his driver model. We have extended his simulation environment to be able to acquire training data for our ANN. The drivers could see the environment via monitor screen and they could control the steering wheel of the vehicle using a cell phone device. An application was developed to send the pitch angle of the phone every 20ms. The rules of table 1 were executing in training phase to record the near and far points and also the time intervals. The angle of steering wheel was sent through the phone. Four graduate students between ages 23 to 32 have been asked to drive in this environment.

The participants were first trained in five 1 minute sessions to get accustomed to the driving simulator. Then they drove the vehicle for five minutes in the simulator at 40, 60, 80, 100 and 120 km per hour speeds (one minute for each speed) and their steering profile were recorded. This data was later used to train the MLP shown in fig.4.

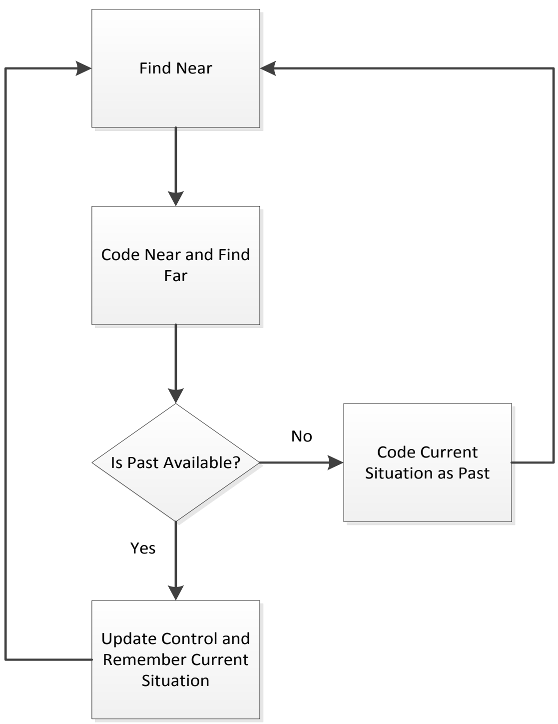

Fig. 5. driver steering model flowchart

TABLE 1. SAMPLES OF TIMES ROMAN TYPE SIZES AND STYLES USED FOR FORMATTING A PES TECHNICAL WORK

Condition	***Action***
Goal is start *Visual buffers are empty* *Imaginal is empty*	*Set goal to code-near* *Request location of near point from where buffer*
Goal is start *Visual buffers are empty* *Imaginal has a world-representation*	*Set goal to code-near* *Request location of near point from where buffer* *Add imaginal contents to declarative memory*
Goal is code-near *Where buffer has near point* *What buffer is empty*	*Set goal to control* *Create a world-representation chunk in imaginal buffer containing* θ_{near} *Request location of far point from where buffer* *Search declarative module for a world-representation*
Goal is control *Where buffer has far point* *Imaginal buffer has* θ_{near} *in a world-representation chunk* *A world-representation chunk exists in retrieval buffer*	*Set goal to remember* *Update world-representation chunk in imaginal buffer to contain* θ_{far} *and current time information* *Request the nature of far point from what buffer* *Invoke the ANN to calculate steering angle change* $\Delta\delta$ *initiate a motor command via manual buffer to change the angle of steering wheel by* $\Delta\delta$
Goal is remember *Imaginal buffer has a world-representation* *What buffer has a chunk representing the far point*	*Set goal to start* *Add the chunk in imaginal buffer to declarative memory*
Goal is control *Where buffer has far point* *Imaginal buffer has* θ_{near} *in a world-representation chunk* *No memory of the past could be retrieved*	*Set goal to start* *Add the information of current situation to declarative memory*

6. COMPARING HUMAN AND MODEL BEHAVIOR

To validate the developed computational model, the comparison between the driver model's behavior and human driver's behavior is required, and since our proposed modelproduces variability in driversbehavior, like human drivers, in this experiment we averaged the performance of human drivers and simulated driver model in 5 trials to achieve more stable results. Human data were from four drivers. The driver model was interacted with the same simulation environment developed in [2] as human drivers used.

Fig. 6 indicates the steering angle used by our models and the drivers while driving at a constant speed of average 90 km per hour.

As shown in these figures our model is quite capable of modeling different driving habits of our participants.

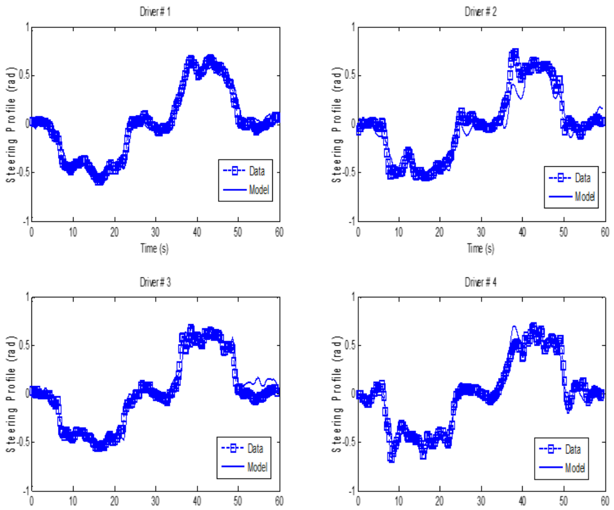

Fig. 6. comparisons between model and drivers at 90 km/h driving

In order to demonstrate the uniformity of prediction's criterions, the results in Fig.7, illustrates the RMSE and R2 of error for various speeds. As shown in this figure, our simulated model has a very good ability in producing human-like behavior and the quality of its predictions does not degrade in various speeds and the model nicely account for the human drivers' behavior.

The results from experiment and simulation indicate that our proposed computational model can perform the process of driver control well and the model's control process is consistent with that of drivers.

7. Conclusion

Knowing the fact that Cognitive Architectures (CAs) are the core of artificial cognitive systems supposedly to specify the human brain at an abstraction level good enough to have a better clue on how to understand theory of mind functioning. The ACT-R cognitive architecture explains the interesting aspects and provides a theory of human attention while driving. At the same time, human attention during driving is a challenging task for the ACT-R cognitive architecture. This work, besides taking advantage of ACT-R cognitive architecture have used MLPs to successfully generate human-like steering behavior. We suggested that using MLPs merely to model automated human skills and utilization of other symbolic ACT-R cognitive architectures for modeling higher level cognitive aspects of human behavior. According to the results, the proposed simulated model has a very good ability in producing human-like behavior and the quality of its predictions does not degrade in various speeds

Our future research aims at further improving the driver computational model by using a more rigorous approach for modeling motor control while driving. This work can also be extended to model other aspects of driving such as lane keeping. The proposed Neuro-ACT cognitive architecture can also be utilized in modeling various human behaviors

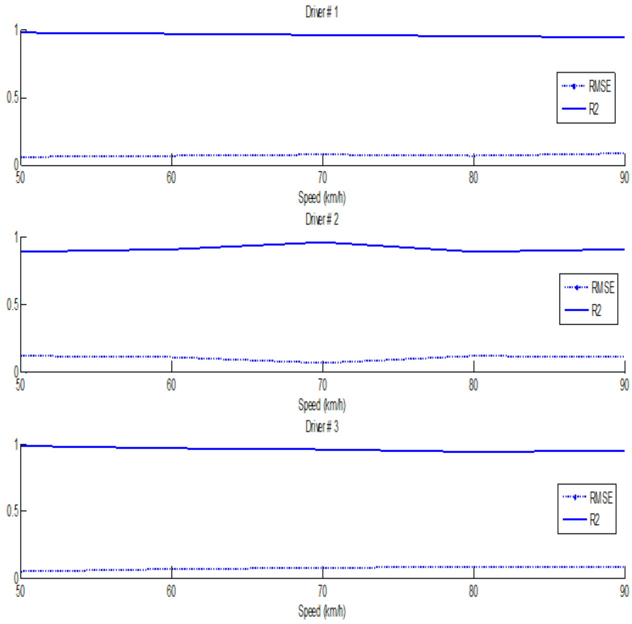

Fig. 7. Goodness of fit criterion

Acknowledgments

We are deeply grateful to QIAU's Incubator Center of Technology Units for providing us the necessary support.

References

[1] D. L. Fisher, M. Rizzo, J. Caird, and J. D. Lee, Handbook of Driving Simulation for Engineering, Medicine, and Psychology. Taylor \& Francis, 2011.

[2] D. D. Salvucci, "Modeling driver behavior in a cognitive architecture," Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 48, no. 2, pp. 362–380, 2006.

[3] D. Fum, F. D. Missier, and A. Stocco, "The cognitive modeling of human behavior: Why a model is (sometimes) better than 10,000 words," Cognitive Systems Research, vol. 8, no. 3, pp. 135–142, 2007.

[4] M. D. Byrne, "ACT-R/PM and menu selection: Applying a cognitive architecture to HCI," International Journal of Human-Computer Studies, vol. 55, no. 1, pp. 41–84, 2001.

[5] P. M. Fitts, "Factors in complex skill training," Training research and education, pp. 177–197, 1965.

[6] J. A. Michon, "A critical view of driver behavior models: what do we know, what should we do?," in Human Behavior and Traffic Safety, Evans, Leonard and Schwing, RichardC., Ed. Springer US, pp. 485–524, 1985.

[7] D. D. Salvucci and N. A. Taatgen, The multitasking mind. Oxford University Press, USA, pp. 70, 2010.

[8] N. A. Taatgen and J. R. Anderson, "Constraints in Cognitive Architectures," The Cambridge handbook of computational psychology, 2008.

[9] Salvucci, D. D., Boer, E. R., and Liu, A., "Toward an integrated model of driver behavior in a cognitive architecture," Transportation Research Record, vol. 1779, pp. 9–16, 2001

[10] Salvucci, D. D., &Macuga, K. L., "Predicting the effects of cellular-phone dialing on driver performance," Cognitive Systems Research, vol. 3, pp. 95–102.

[11] Salvucci, D. D., Liu, A. ,and Boer, E. R., "Control and monitoring during lane changes," Vision in Vehicles, vol. 9, 2001.

[12] Byrne, M. D., "ACT-R/PM and menu selection: Applying a cognitive architecture to HCI," International Journal of Human Computer Studies, vol. 55, pp. 41–84, 2001.

[13] Wilkie, R. M., and Wann, J. P., "Controlling steering and judging heading: Retinal flow, visual direction and extra-retinal information," Journal of Experimental Psychology: Human Perception and Performance, vol. 29, pp. 363–378, 2003.

[14] Matessa M., "An ACT-R Modeling Framework for Interleaving Templates of Human Behavior," Proc. of the Twenty-sixth Annual Conference of the Cognitive Science Society, Chicago, IL., pp. 903-908, 2004.

[15] Liu Y. and Wu Z., "Comfortable Driver Behavior Modeling for Car Following of Pervasive Computing Environment," International Conference on Computational Science (3), pp. 1068-1071, 2005.

[16] Liu Y. and Wu Z., "Driver Behavior Modeling in ACT-R Cognitive Architecture," Journal of Zhejiang University 2006.

[17] Liu Y. and Wu Z., "Multitasking Driver Cognitive Behavior Modeling," Prof. of 3rd IEEE Conference on Intelligent Systems, London, 4-6 September, 2006.

[18] Liu Y. and Wu Z., "Urgent Driver Behavior Modeling in Cognitive Architecture," Proc. of 18th ICTCT Workshop, Helsinki, 27-28 October, 2005.

[19] Liu Y. and Wu Z., "A Smart Car Control Model for Driver's Comfort of Car Following," Proc. of IEEE Intelligent Vehicles Symposium, Las Vegas, pp. 833- 839, 2005.

[20] A. Mihalyi, B. Deml, and T. Augustin, "A contribution to integrated driver modeling: A coherent framework for modeling both non-routine and routine elements of the driving task," in Digital Human Modeling, Springer, pp. 433–442, 2009.

[21] J. A. Bargh, K. L. Schwader, S. E. Hailey, R. L. Dyer, and E. J. Boothby, "Automaticity in social-cognitive processes," Trends in cognitive sciences, 2012.

[22] D. D. Salvucci and R. Gray, "A two-point visual control model of steering," Perception-London, vol. 33, no. 10, pp. 1233–1248, 2004.

[23] P. M. Fitts, "Factors in complex skill training," Training research and education, pp. 177–197, 1965.

[24] J. A. Michon, "A critical view of driver behavior models: what do we know, what should we do?," in Human Behavior and Traffic Safety, Evans, Leonard and Schwing, RichardC., Ed. Springer US, pp. 485–524, 1985.

[25] Land, M., and Horwood, J., "Which part of the road guide steering?," Nature, vol. 3, no. 77, pp. 339-340, 1995.

[26] Liu, Y., "Queuing network modeling of elementary mental processes," Psychological Review, vol. 103, pp. 116-136, 1996.

[27] Pomerlau, D., and Jochem T., "Rapidly adapting machine vision for automated vehicle steering," IEEE Expert, vol. 112, no. 19-27, 1996.

[28] C. Wu, and Y. Liu, "Queuing network modeling of the Psychological Refractory Period (PRP)," Psychological Review, vol. 115, no. 4, pp. 913-954, 2008.

[29] D. T. McRuer, R. W. Allen, D. H. Weir, and R. H. Klein, "New results in driver steering control models," Human Factors, vol. 19, no. 4, pp. 381-397, 1977.

[30] M. Kondo and A. Ajimine, "Driver's sight point and dynamics of the driver-vehicle system related to it," Proceedings of the SAE automotive engineering congress, Detroit, MI, 1968.