COMPUTATIONAL MODELING OF DRIVER LATERAL CONTROL ON CURVED ROADS WITH INTEGRATION OF VEHICLE DYNAMICS AND REFERENCE TRAJECTORY TRACKING

Heejin Jeong[1], Fred Feng[2], and Yili Liu[1]
[1]University of Michigan
[2] University of Michigan Transportation Research Institute
Ann Arbor, Michigan, USA
Email: heejinj@umich.edu

Summary: Driver's lateral control on curved roads plays a significant role in reducing or avoiding the crashes. To understand and predict driver performance on curved roads, a computational model was developed in a cognitive architecture, the Queueing Network-Model Human Processor (QN-MHP), with the integration of vehicle dynamics principles (i.e., how to steer based on near and far angles) and the reference trajectory tracking method (i.e., how to steer on the road varying with radius of road curvature). The model was implemented with four major components: road information, vehicle dynamics, visual perception, and cognition & motor controls. The model outputs were validated with the corresponding human subject performance in the literature. The performance results of the model highly fitted the human subject data such as steering wheel angle.

INTRODUCTION

Nearly 25 percent of drivers who die each year on the American roadways are due to fatal crashes on curved roads (McGee & Hanscom, 2006). Therefore, it is important to study and understand driver lateral (or steering) control on curved roads in order to help reduce or avoid the crashes. In this regard, computational modeling of driver lateral control on curves can play a significant role in supporting quantitative analysis of driver's performance.

Since the last few decades, a variety of modeling studies have been conducted to quantify driver behavior in lateral control. One method used was based on control theory that assumes the human driver as one of the control elements in the driver-vehicle system (e.g., MacAdam, 2003). Another method used for the lateral control modeling was the driver preview model based on imitating drivers' preview/predictive behaviors (e.g., Ungoren & Peng, 2005). Researchers have also started to model driver performance using task-independent cognitive architectures, based on experimental psychology and neuroscience findings. Examples of cognitive architecture based driver model include the Adaptive Control of Thoughts-Rational (ACT-R) (e.g., Salvucci, 2006), and the QN-MHP (e.g., Liu, Feyen, & Tsimhoni, 2006).

In this study, we used the QN-MHP cognitive architecture to model driver lateral control on curved roads. The QN-MHP is a simulation model of cognitive processing system based on the queueing network theory of human performance (see Liu et al., 2006 for more details). The QN-MHP architecture is composed of three subnetworks (perceptual, cognitive, and motor) and each subnetwork consists of multiple servers representing the functional components of the brain and body for human performance. The servers are connected by routes, while entities travel through the routes. One of the merits of using the QN-MHP is that it allows more than one server to act either in parallel or in serial. Thus, it is possible to model human performance in multi-task scenarios represented as multiple flows of entities, such as the driving performance under multi-task conditions.

Since the QN-MHP was first developed a few decades ago, it has been successfully used to model a wide range of human performance, including transcription typing (Wu & Liu, 2008), visual search (Lim, Liu, & Tsimhoni, 2010), and swiping on touchscreens (Jeong & Liu, 2016). In this study, a novel method, called reference trajectory tracking, was used to control vehicle's lateral motion in order to obtain higher accuracy of modeling both the position and time elements. This method has been originally used to design autonomous vehicle's lateral movement by minimizing the spatial and temporal errors from the reference trajectory (e.g., Aguiar & Hespanha, 2007; Talj, Tagne, & Charara, 2013). Using the reference trajectory tracking concept (i.e., by making the virtual vehicle follow the built-in reference trajectory), we have developed a model to simulate vehicle's lateral movement on curved roads with multiple levels of radius of road curvature. After model development for driver's lateral control, model validation was conducted with the existing experimental data from Tsimhoni & Green (2003), with lateral control measurements such as steering wheel angle.

COMPUTATIONAL MODELING OF LATERAL CONTROL

The model built in this study combines the original QN-MHP architecture with the road curvature information and vehicle dynamics for modeling driver's lateral control in driving on curved roads (See Figure 1). The lateral control model is implemented in MATLAB-Simulink and has four main components: (1) road information, (2) vehicle dynamics, (3) visual perception, and (4) cognition & motor controls. As shown in Figure 1, entities carrying road information enter the QN-MHP architecture as well as the vehicle dynamics component:

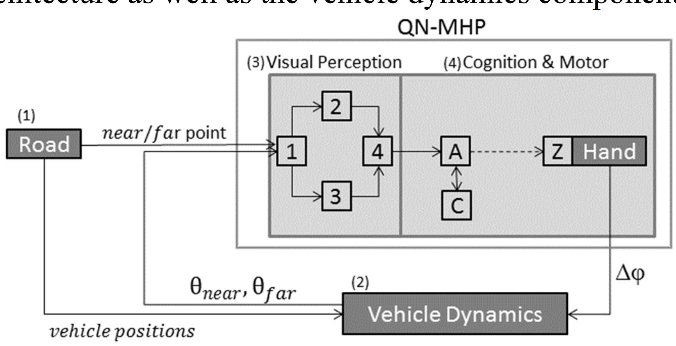


Figure 1. Structure of four components with the QN-MHP

At the QN-MHP architecture, first, the entities enter the visual perception subnetwork: Servers 1 (visual input) → 2/3 (visual recognition/localization) → 4 (visual integration). The entities collected in Server 4 move to the cognitive & motor subnetworks by entering through Server A (visuospatial sketchpad) and exiting through Server Z (Actuators; connected to the hand server in this study). Once the entities arrive at the hand server, a signal is sent to the vehicle dynamics component so that the component can prepare the settings for carrying out the lateral control (e.g., yaw angle adjustment and near/far angle acquisition). At Server C (central executor), driver's steering wheel angle (φ) is determined (by Equation 5), based on the near/far angle obtained from the vehicle dynamics component. In this study, Server C was used (rather than the combination with Server F) to perform the driving lateral control because it was assumed that adjusting the steer wheel on curved road having a constant radius of curvature is a simple cognitive process, compared to such complex cognitive activities, mostly performed through Server F (e.g., multiple-choice comparison and decision, and math calculation).

On the other hand, the entities that travel from the road information component to the vehicle dynamics component carry road information (e.g., radius of road curvature and 2-dimentional coordinates of vehicle's reference trajectory), which are used to predict the yaw angle and vehicle's position, then eventually near angle and far angle (by Equations 3 and 4) at the vehicle dynamics component. The output (φ) determined in Server C is used for driver's lateral control when the entities arrive at the vehicle dynamic component in the next cycle after circulating through Server A and Server Z (and the hand server).

Road Information Component

One of the major factors that affect driver's visual perception on the curved road is the radius of road curvature (Dickmanns & Zapp, 1987; Shinar, Rockwell, & Malechi, 1980). In geometry, the radius of curvature, *R*, at a particular point is defined as the radius of the most approximate circle touching the point. With the assumption that the curve is differentiable up to the second order, the formula for the radius of curvature at any point *x* for the curve *z* = f(*x*) is given by:

$$R = \left| \frac{(1+(\frac{dz}{dx})^2)^{3/2}}{\frac{d^2z}{dx^2}} \right| \tag{1}$$

, where *x* is the lateral coordinate and *z* is the longitudinal coordinate (Do Carmo, 1976).

Although this radius varies as the vehicle moves along the curve (Hastie & Stuetzle, 1989), it was assumed that the radius of road curvature in the current model indicates the radius when the vehicle is at the middle of curve. The road curvature parameter (*R*) is used as an input to estimate the yaw angle (ψ) at the vehicle dynamics component. The built-in reference trajectory data are used to determine the 2-dimensional geometrical center of vehicle position at the vehicle dynamics component. The details are described in the vehicle dynamics component section.

Vehicle Dynamics Component

In this component, three vehicle dynamics measurements are determined: vehicle positions, yaw angles, and near/far angles. The vehicle positions (i.e., lateral and longitudinal positions) are determined by minimizing the time/space-based errors from the built-in reference trajectory data, using interpolation method. Since the reference trajectory data are designated every 100 msec and the current simulation cycle time is every 50 msec, it is necessary to find the closest cycle time of reference trajectory; the trajectory data at the closest cycle time are regarded as the current vehicle positions.

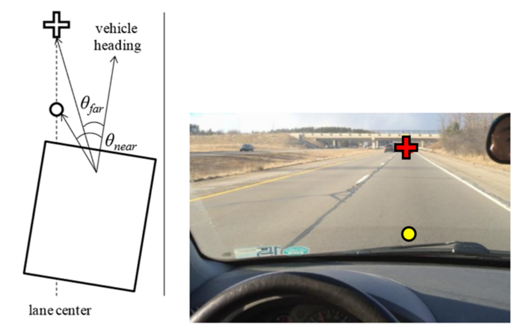
In parallel, yaw angles are determined by the simulation cycle time. Yaw motion is one of the significant elements for controlling lateral movement in vehicle dynamics (Ackermann & Bünte, 1997; Rajamani, 2012). Yaw angle (ψ) is defined as the angle between the direction of the vehicle heading and the direction of the lane center. In the current study, the yaw angle was determined with the finding from Rajamani (2012):

$$\psi_t - \psi_{t-1} = \frac{v_t \cdot \Delta t}{R} \tag{2}$$

v_t is the vehicle speed at time t, whereas R is radius of road curvature. Δt is the time elapsed from last cycle. Each cycle time depends on two time components: inter-arrival time and server processing time. The inter-arrival time is a fixed time, currently set as 50 msec as the default value for the visual stimulus generation rate. The server processing time is set as a shifted exponential distribution with mean a and an axis shift *b*. It is written as *E(a) + b*: the values of parameter <*a, b*> in this study were set as <17, 25> for perceptual servers, <12, 6> for cognitive servers, and <14, 10> for motor servers, based upon Feyen (2002). The present model steers the vehicle at a fixed speed of 72 km/h (or 45 mph). Using the lateral vehicle position (X_t) and yaw angle (ψ_t) at each cycle time, near/far angles (i.e., the results of Equations 3 and 4) are determined at this vehicle dynamics component.

Visual Perception Component

In the current model, the near point and far point were generated every 50 msec and they were used as visual stimuli inputs at the visual perceptual servers (Servers 1 - 4). The near point indicates a visible point in front of the vehicle that the driver uses for estimating how adjacent the vehicle is to the center of lane, whereas the far point represents a visible point in front of vehicle that the driver uses to estimate a near future position (See Figure 2).



(a) Bird's-eye view (b) Driver's point-of-view

Figure 2. (a) Near/far angles and (b) near/far points (a yellow dot and a red cross, respectively) (Feng, 2015)

Figure 2 illustrates the near angle and far angle: the near angle refers to the direction from the vehicle pointing to the near point relative to the direction of the vehicle heading, while the far angle represents the direction from the vehicle pointing to the far point relative to the direction of the vehicle heading.

The near angle and the far angle are determined (by using Equations 3 and 4 below) at the vehicle dynamics component after the input data for the component are obtained, such as the lateral coordinates of vehicle position (X_t) and yaw angles (ψ_t). While *nearDistance* indicates the near point location (in distance) on the road (set as a constant, 10 m), *farTHW* represents the far point location (in time headway) on the road (set as a constant, 4 s).

$$\theta_{near,t} = \tan^{-1}\left(\frac{X_t}{nearDistance}\right) + \psi_t \tag{3}$$

$$\theta_{far,t} = \tan^{-1}\left(\frac{X_t}{farTHW \cdot v_t}\right) + \psi_t \tag{4}$$

Cognition and Motor Controls Component

Once the near and far angles are stored in the cognitive subnetwork, steering wheel angle adjustment is conducted at Server C. The present model uses a steering wheel angle (φ) equation formulated by Salvucci (2006).

$$\Delta\varphi = k_{far} \cdot \Delta\theta_{far} + k_{near} \cdot \Delta\theta_{near} + k_l \cdot \min(\theta_{near} - \theta_{near_{max}}) \cdot \Delta t \quad (5)$$

In which: $\theta_{near_{max}}$ (set as a constant, 0.07 radian) is for limiting the contribution of the θ_{near} to changes in steering wheel angle. k_{far}, k_{near}, and k_l indicate the weights for the three terms (set as 7, 4, and 3, after the multiple validations to obtain similar results with the experimental data). The motor server (indicated as Z in Figure 1) sends a signal to the hand server so turning actions are performed in the vehicle dynamic component using the data of the change of steering wheel angle (Δφ) in each cycle. The turning action time (i.e., steering time), is taken, based on the estimation with a steering rate of 963 degree/sec from Forkenbrock & Elsasser (2005).

VALIDATION RESULTS

The lateral control driving model was evaluated using the empirical data of Tsimhoni & Green (2003), conducting a driving simulation experiment with 24 participants (12 younger (M = 23, SD = unknown); 12 older (M = 68, SD = unknown)). They drove on a 3.6 m wide single-lane road made up of a curve (two levels: R = 200 and 400 m) connecting two straight roads back and forth of the curved road. The data collected when subjects drove on curved roads were used for validation of the current model. The participants were asked to drive with a constant cruise controlled speed (72 km/h or 45 mph).

Figure 3-(a) shows the mean yaw angles generated by the vehicle dynamics module (in degree; during 10 runs of simulation) in driving on four different radii of road curvature. The mean yaw angles are fairly constant over time, and the value decreases as the vehicle is driven on a curve with a larger radius. Similarly, the mean steering wheel angle (in degree; during 50-second driving) decreases as the driver model drove on larger-radius curves (See Figure 3-(b)).

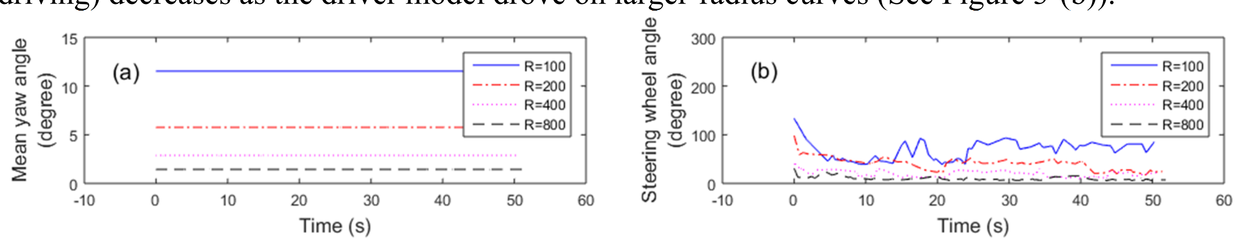


Figure 3. (a) mean yaw angle and (b) steering wheel angle of simulation results (in degree) in four different radii of road curvature at 72 km/h

With regard to the comparison between simulation and experimental results, as shown in Figure 4, the mean steering wheel angles of the driver model quite closely fit the experimental data for both radii of road curvature (R = 200 and 400 m).

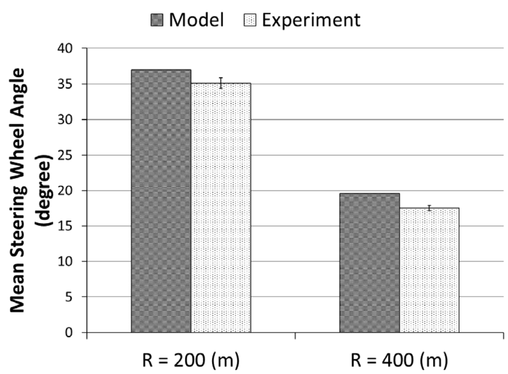


Figure 4. Mean steering wheel angle comparisons between the QN simulation and experimental results

DISCUSSION

Using the vehicle dynamics and the method of reference trajectory tracking, as seen in Figure 4, the model simulation was able to generate mean steering wheel angle data that were fairly similar to the human experimental study (mean estimation error = 8.6 %, RMS = 1.98). Furthermore, since the vehicle in simulation was to be followed the built-in reference trajectory, the lateral gap between the center of vehicle and road was close to zero.

In Tsimhoni & Liu (2003), a driver steering model was also successfully developed using processing logics including detecting orientation, selecting a steering strategy, and steering action: the model yielded the steering angle and lateral position in two fixed radii of curvature, similar to the experimental data using human subjects. However, one major contribution of this current study is that the method of reference trajectory tracking can help prediction of the driving performance on different curved conditions (rather than just fixed radius of curvature). In other words, once having any built-in reference trajectory data (including vehicle coordinates at each cycle time), it will be possible to model the vehicle control on that trajectory.

Some limitations of the current study are that: (1) the current lateral vehicle control model ran with a fixed speed; (2) only the mean steering wheel angle measurement has been validated due to the lack of corresponding human experimental data for validation. We are extending the present model so that it can run in more complicated settings, including a longitudinal speed control, with or without a lead vehicle, and so forth. Moreover, using the reference trajectory tracking method which can demonstrate vehicle's movement in three-dimensional space (Aguiar & Hespanha, 2007), we plan to model driver elevation control on the upward/downward slopes.

REFERENCES

Ackermann, J., & Bünte, T. (1997). Yaw disturbance attenuation by robust decoupling of car steering. *Control Engineering Practice*, *5*(8), 1131-1136.

Aguiar, A.P. and Hespanha, J. P. (2007). Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Transactions on Automatic Control*, *52*(8), 1362-1379.

Dickmanns, E. D., & Zapp, A. (1987, February). A curvature-based scheme for improving road vehicle guidance by computer vision. *In Cambridge Symposium_Intelligent Robotics Systems* (pp. 161-168). International Society for Optics and Photonics.

Do Carmo, M. P. (1976). *Differential geometry of curves and surfaces* (Vol. 2). Englewood Cliffs: Prentice-hall.

Feng, R. (2015). *Queuing Network Modeling of Human Multitask Performance and its Application to Usability Testing of In-Vehicle Infotainment Systems* (Doctoral dissertation, University of Michigan).

Feyen, R. G. (2002). *Modeling human performance using the queuing network-model human processor (QN-MHP)* (Doctoral dissertation, University of Michigan).

Forkenbrock, G. J., & Elsasser, D. (2005). An assessment of human driver steering capability. *National Highway Traffic Safety Administration DOT HS* 809, 875.

Hastie, T., & Stuetzle, W. 1989. Principal curves. *Journal of the American Statistical Association*, *84*(406), 502-516.

Jeong, H., & Liu, Y. (2016, September). Computational Modeling of Finger Swipe Gestures on Touchscreen Application of Fitts' Law in 3D Space. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 60, No. 1, pp. 1721-1725). SAGE Publications.

Lim, J. H., Liu, Y., & Tsimhoni, O. 2010. Investigation of driver performance with night-vision and pedestrian-detection systems—Part 2: Queuing network human performance modeling. *IEEE Transactions on Intelligent Transportation Systems*, *11*(4), 765-772.

Liu, Y., Feyen, R., & Tsimhoni, O. (2006). Queueing Network-Model Human Processor (QN-MHP): A computational architecture for multitask performance in human-machine systems. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *13*(1), 37-70.

MacAdam, C. C. (2003). Understanding and modeling the human driver. *Vehicle System Dynamics*, *40*(1-3), 101-134.

McGee, H. W., & Hanscom, F. R. (2006). *Low-cost treatments for horizontal curve safety* (No. FHWA-SA-07-002).

Rajamani, R. (2012). Lateral vehicle dynamics. *In Vehicle Dynamics and control* (pp. 15-46). Springer US.

Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors: The Journal of the Human Factors and Ergonomics Society*,*48*(2), 362-380.

Shinar, D., Rockwell, T. H., & Malechi, J. A. (1980). The effects of changes in driver perception on rural curve negotiation∗. *Ergonomics, 23(*3), 263-275.

Talj, R., Tagne, G., & Charara, A. (2013, July). Immersion and invariance control for lateral dynamics of autonomous vehicles, with experimental validation. In *European Control Conference (ECC'13)* (pp. 968-973).

Tsimhoni, O., & Green, P. (2003). Time-sharing of a visual in-vehicle task while driving: The effects of four key constructs. *In Proceedings of the Second International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design. Park City, Utah. July* (pp. 21-24).

Tsimhoni, O., & Liu, Y. (2003, October). Modeling Steering Using the Queueing Network—Model Human Processor (QN-MHP). *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 47, No. 16, pp. 1875-1879). SAGE Publications.

Ungoren, A. Y., & Peng, H. (2005). An adaptive lateral preview driver model. *Vehicle system dynamics, 43*(4), 245-259.

Wu, C., & Liu, Y. 2008. Queuing network modeling of transcription typing. *ACM Transactions on Computer-Human Interaction (TOCHI), 15*(1), 6.