System Identification Design of a Mathematical Model for an Autonomous Vehicle

Emily Sheetz Timothy Tadros Nathalie Risso Jonathan Sprinkle

Abstract—**An important part of building an autonomous mobile control system is creating a model that accurately reflects the system's behavior in order to predict and plan the future state of the system. Several approaches to building a model of an autonomous vehicle have been implemented and studied. However, the trade-off between the accuracy and computation time of a model makes it difficult for an autonomous system to use accurate models in real-time to plan its trajectory. In the research presented in this paper, we use system identification to develop a model that accurately predicts the trajectory of the vehicle while reducing the computation time of the model. This model is built from experimental data collected from the Cognitive Autonomous Test (CAT) Vehicle at the University of Arizona. Our model is implemented on a hybrid predictive controller and tested in simulations and real-world applications. The controller uses the model to follow a planned trajectory and avoid obstacles in the state space with reasonable computation time. While the proposed model is specific to a particular autonomous vehicle, our methods and models could be applied to other autonomous systems.**

I. Introduction

A. Background and Motivation

The technology of autonomous vehicles is rapidly growing in quantity and quality. For autonomous systems to be safe and predictable, the controller that manages the vehicle's state commonly uses sensor information along with a model of the system. If a model accurately describes the behavior of an autonomous system, it can aid the tasks of path-planning and trajectory-following. Path-planning consists of dynamically designing a valid trajectory to follow, as the vehicle learns about its environment through sensor data, GPS information, or camera technologies. Trajectory-following occurs along small time-steps as the controller compares feedback error or predictions from the model with the actual state of the vehicle and corrects the vehicle's state variables to more closely follow the desired trajectory.

In the particular case of predictive controller, such as MPC and DMC, the system requires accurate predictions from the model to make decisions in real-time, so even in novel environments, the model must be computationally efficient. If the model accurately describes the behavior of the car but cannot be computed in a reasonable time, then it cannot safely be used for the purposes of path-planning and trajectory-following. The model must enable the system to make accurate real-time decisions in order to avoid obstacles and maintain a safe trajectory.

E. Sheetz is a junior at the Department of Mathematics and Computer Science, Monmouth College, Monmouth, IL 61462, USA. Email: esheetz@monmouthcollege.edu

T. Tadros is a senior at the Department of Computer Science, Dartmouth College, Hanover, New Hampshire, USA. Email: timothy.m.tadros.17@dartmouth.edu

N. Risso is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85719, USA. Email: nrisso@email.arizona.edu

J. Sprinkle is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85719, USA. Email: sprinkle@ece.arizona.edu

B. State of the Art

To achieve an acceptable control of an autonomous vehicle, a model that generates accurate predictions in a reasonable timeframe is required. There are several challenges present in building such a model of an autonomous vehicle that have been addressed in the literature on previous works with autonomous systems.

One challenge is associated to measuring information about the autonomous system and its environment. Using either cameras, GPS measurements, or external sensors, the autonomous system can learn about its environment and its current state to choose a path that is safe and efficient. However, the sampling rates of these sensors limit how often the system can receive these data and respond accordingly. GPS is known to have a low sampling rate [19], which means that the autonomous system may not have enough information between data samples to stay on its path or avoid obstacles. One way to get around this is to integrate GPS with other sampling methods, such as laser sensors [5] [12] [14]. Although these sensors can be more efficient, they are expensive and subject to interference from other laser sensors, and therefore not a good universal option for autonomous vehicles [4] [9]. On the other hand, cameras have a high sampling-rate but are ineffective in night-time, foggy, or bright environmental conditions [4]. Computer vision can be used to achieve strong results in poor conditions, but most of these algorithms offer more unique approaches than their daytime counterparts, so they are often complex [6] [9].

The next challenge of modeling to be addressed is the type of model to use. The model should be sophisticated enough to provide useful information about the system to the controller, but simple enough that it does not create a large computational burden. Linear models come with low computation time, but are rare in real-world applications [17]. However, nonlinear models that more accurately fit real-world processes come with significantly greater computation time. Kinematic models are easier to design and implement because they give a coarse understanding of the system, and as a result can be computed quickly. Dynamic models take significantly longer to compute, but provide more detailed and accurate information about the state of the autonomous vehicle. Hybrid models incorporate the strengths of both kinematic and dynamic models while compensating for their weaknesses [18]. Hybrid models switch between discrete-state and continuous-state models depending on how much information is required to maintain safe operation of the autonomous vehicle. Switching logic determines the conditions under which each model should be used.

Another challenge in building a model of an autonomous vehicle exists in choosing and evaluating the parameters to be used in the model. Parameters can be assigned values based on the properties of the car [15] [16] [18] or based on parameter estimation [1] [7]. The former approach risks assigning inaccurate values to the parameters, which may affect the reliability of the model. For example, some parameter values may vary based on environmental factors such as temperature or tire pressure. Thus, assigning values to these parameters will be less accurate than estimating the values of the parameters based on some objective function.

Overall performance of the autonomous system behavior is defined by its control strategy. Several approaches can be considered here. Batch controllers do not utilize mathematical models; in the case of an autonomous vehicle, a batch controller may break driving down into motion primitives and control the vehicle through combinations of motion primitives [10]. Proportional Integral Derivative (PID) controllers are often too simple for nonlinear systems. However, if utilized with another controller, PID controllers can manage low-level control tasks while the other controller manages higher-level tasks [12]. Model predictive control (MPC) is widely used because closed-loop predictive controllers can utilize feedback to effectively handle system constraints and disturbances [2] [11]. However, nonlinear models implemented in model predictive controllers often come with heavy computational burden. Hybrid controllers model continuous state processes with models that are computed in discrete time [3] [18]. Artificial intelligence (AI) based path-planning can plan and follow trajectories accurately and safely even in unfamiliar environments with stationary and moving obstacles [8]. However, these controllers are complex and computationally time consuming. Artificial neural network approaches are just as effective in path-planning and trajectory-following, but require exhaustive understanding of the environment and possible scenarios the network could encounter [12]. Furthermore, neural network approaches utilize fuzzy logic, which may be too imprecise in situations when safety is a priority.

C. Contributions

In this paper, we use system identification to develop a model that accurately represents the current state of the CAT Vehicle and predicts the future states of the system under a range of typical driving conditions. In particular, our contributions include:

- Propose an experimental model that outperforms predictions obtained using kinematic and dynamic models for the CAT Vehicle behavior. Our model also provides reduced computation time associated with the model prediction, suitable to be used on model-based optimal controllers.
- Redefined switching logic based on experimental analysis improves computational time and state predictions.

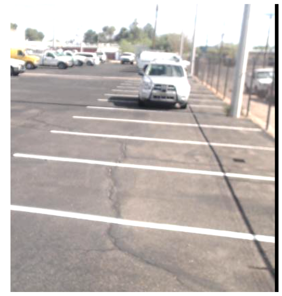

Fig. 1. Example of the straight line GPS sensor characterization experiments. Parking lanes were used as references for the car's true position.

D. Organization of the Paper

This paper is organized as follows. Section II presents experimental design for data collection, methods for data processing, and development of the model. Simulated tests and real-world test results are presented in Section **??**.

II. Methods

System identification is the process of using experimental data to characterize a system's behavior [13]. We use system identification methods to develop our model of the CAT Vehicle by designing experiments, gathering and analyzing data, finding and validating the model.

Because our experiments were designed to be representative of typical driving tasks, the model developed from the data is accurate within the tested ranges of velocity, steering angle, and acceleration.

A. GPS Characterization

The GPS sensor on the CAT Vehicle–a NovAtel IMU-CPT Global Navigation Satellite System (GNSS)–takes noisy measurements and, according to the manufacturer, has an accuracy of 1.2 meters. The raw GPS data are not precise enough for the development of an accurate model for the vehicle and need to be improved.

Several approaches to improving GPS accuracy were considered. Computer vision utilizes cameras to determine the approximate true position of the vehicle, which can be used to verify the location of the vehicle according to the GPS measures. Similarly, readings from multiple other sensors, such as lasers, in addition to GPS could verify the position of the vehicle. While the use of cameras or lasers could improve the accuracy of GPS measurements by verifying the position of the vehicle, these other sensors have error as well. This means that the true position of the vehicle determined by the other sensors is actually inaccurate.

We designed a set of experiments that provide data to compare measures from the GPS sensor to the car's true position, estimated obtained through computer vision. The purpose of these experiments is to characterize the inherent error of the GPS sensor, develop a function or filter to eliminate noise and improve accuracy of the measures, and ultimately obtain more accurate position information, to be used for the development of the model for the vehicle.

The experiments to characterize GPS error included driving along a straight line and making right and left turns in a parking lot. These experiments were designed to represent some typical driving conditions, but were ultimately limited by the size of the testing area. Straight lines were tested at 2.5 and 5 meters per second, while right and left turns were tested at 2.5 meters per second.

The lanes in the parking lot were used as way-points–stationary points with known position, estimated by multiple GPS tools–to measure when the vehicle reached a certain point, as seen in Figure 1. The true position of the vehicle was determined using information from videos recorded by a Logitech HD Pro C290 Webcam at a frame rate of 30 frames per second. Computer vision tools were used to see when the vehicle crossed the way-points along its path (See Fig. 1). The GPS sensor measurements were compared to the known position of the vehicle according to the video to measure the error in the GPS sensor.

Once we had finished gathering data from our experiments, we examined the frequency of the signal from the GPS using spectral analysis. We developed a lowpass filter to remove high frequency changes in the GPS measures and to smooth the GPS data. The smoothed data could then be compared to the true position of the vehicle.

The developed filter was used to correct the GPS measures offline after experiments for model development were run.

B. Experiment Design for Model Development

To develop our model, we designed experiments to test how the CAT Vehicle operates under ranges of typical driving conditions. The vehicle collects data on velocity, steering angle, acceleration, and brakes every 0.02 seconds. The NovAtel IMU-CPT GNSS collects position information every 0.01 seconds. The sampling rate of the vehicle–0.02 seconds–determined the sampling rate for our experiments.

The experiments test dynamical behavior on typical driving ranges. Velocity was tested from 2.5 to 10 meters per second. Steering angle is recorded by the car as a percent from -100 to 100. Accelerator and brakes are recorded as a percent from 0 to 100. These percentages will later be converted to real values after experiments are conducted.

Experiments were put into categories: straight paths, turns, and circles. For the straight path experiments, we commanded a velocity (corresponding to the range above, tested in increments of 2.5 meters per second) to the car and allowed the car to move in a straight path for approximately 1 minute or 3000 data samples. On a few of the experiments, we manually accelerated and decelerated once the vehicle reached a predetermined velocity. For turn experiments, we maintained an approximately constant velocity and steering angle of the car, collecting data for 1 minute as before. For circle experiments, we drove the vehicle in as tight a circle as possible, testing both negative (left) and positive (right) angles.

Each test was run ten times. Repetition of the experiments provides sufficient data to characterize random error in the measures collected from the vehicle, the GPS, and the controller that manages velocity once given a commanded velocity.

A longer test designed to last 5 minutes, containing about 150,000 data samples was obtained to perform cross validation on the model.

C. Pre-processing the Data

After collecting the data from the experiments, a number of steps to preprocess the data are necessary. We first needed to make sure measurements for each variable were taken at the same timestamps. Since the sensors take measurements independently, the measurements we collected were taken at different timestamps. Furthermore, the samples were not taken at uniform time intervals, and test space often limited the number of samples that could be taken during an experiment. Interpolating the data gathered so that each experiment contained 3000 uniformly spaced samples resolved these issues. We created a time vector based on the difference between the start (t_s) and end times (t_e) of each experiments and the number of samples desired for each experiment (in this case a constant 3000).

$$t(k) = (k-1) \cdot ((t_e - t_s)/3000) + t_s \tag{1}$$

This time vector was used to interpolate the results collected from the experiment. Mean errors produced by interpolating were all in the range of 0 to 0.05.

After interpolating the data, each signal measurement is filtered using a $20th$ order FIR low pass filter and smoothed using a moving average filter with a span of 10 elements.

The next step in processing the data is converting the steering percentages into real values. For steering information we consider steering of the wheels can be estimated using Ackerman's steering model and a linear relationship between the steering percentage and angle. As specified by the manufacturer, the minimum radius of curvature of the car is 5.58 meters. We assume this is equivalent to a steering percentage of 100. The maximum steering percentage is equivalent to a steering angle of 35.5 degrees. We convert the measured steering percentage into a steering angle of an imaginary center tire by the equation $\delta = 35.5/100 \cdot p_m$, where p_m is the measured steering angle percent and δ is the measured angle of an imaginary center tire. This is converted into a radius of curvature using the following equation specified by the vehicle's manual:

$$r_m = \frac{\tan \dfrac{\delta}{1 + 0.0015v^2}}{L} \tag{2}$$

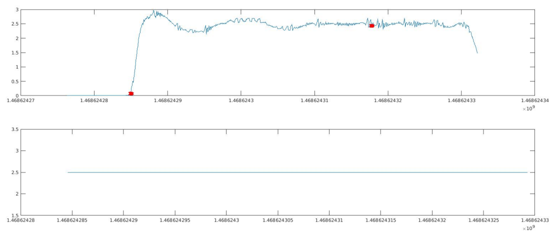

Fig. 2. Calculation of delay and settling time. The red dots show when the car starts moving (dot 1) and when the car reaches an equilibrium point (dot 2).

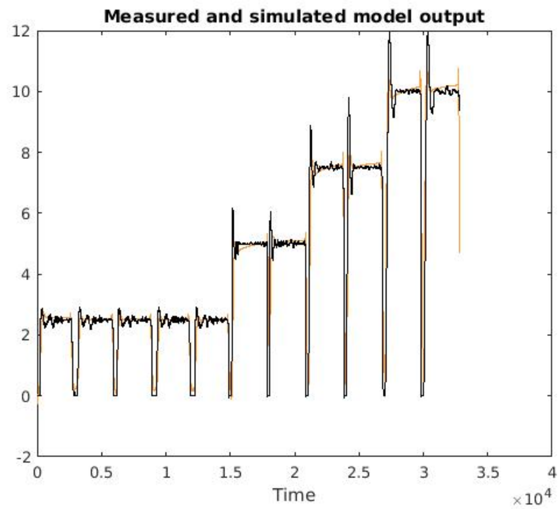

Fig. 3. Internal Controller Dynamics model shown on the validation set (half of the experiments run)

where v is the velocity and L is the wheel base of the car. Then, the measured radius of curvature is converted to steering angles using the Ackerman steering model:

$$\delta_i = \arctan \frac{L}{r_m - \dfrac{T}{2}} \tag{3}$$

$$\delta_i = \arctan \frac{L}{r_m + \dfrac{T}{2}} \tag{4}$$

where T is the tread, and δ_i and δ_o are the inner and outer wheel steering angles respectively.

After pre-processing the data, the data were organized into matrices. An example matrix is shown in Table II-C.

vel	input vel	accel	δ steer	brake	lat	long
0.07	0	0	-3.356	0	32.3	-110.9
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

After creating the matrices, the delay and settling time of the internal controller for the velocity were measured. The mean settling time of the experiments was 30.11 seconds and the mean delay of the system was 0.7387 seconds, as computed using the graph in Fig. II-C.

D. Model Development

Model development was separated into characterization of internal controller dynamics and dynamical car model development.

1) Internal controller dynamics: Given that the autonomous vehicle considered in this study runs XXX ADD SOFTWARE VERSION HERE, an internal controller is provided by the manufacturer for which no characterization was available. This controller generates additional dynamics to the CAT Vehicle response. In order to characterize the behavior of this internal controller two sets of data were considered: velocity profile and wrench control. Experimental setting consisted of 25 experiments where a velocity command was input into the car and the actual velocity.Experimental data was fitted to a continuous time transfer function model of order 3, with a 0.42 second delay[1].

$$h_1(s) = e^{-0.42s} \frac{0.03679s^2 - 0.1981s + 0.7647}{s^2 + 1.629s + 0.7522} \tag{5}$$

Identification results are presented below.

Data	Fit Percent
Test Set	78.53
Training Set	74.53

Figure 3 presents representative cross validation error with respect to the best fit model, selected base on model simplicity and prediction error.

2) Car Dynamics model: . For the development of this model, we used the MATLAB System Identification toolbox to test how different models would fit our data. We began by estimating the relevance of parameters measured by the CAT Vehicle and parameters computed after experiments were run. Computed parameters included the steering angle measured in radians, the cosine, sine, and tangent of the steering angle, and the position of the vehicle in meters. Correlation analysis and least squares estimation equation [13] were used to define a identification matrix from the original 17 measured variables considered as

$$\hat{\Theta} = (M_i^T M_i)^{-1} M_i^T Y \tag{6}$$

Where M_i initial measurement matrix, Θ estimated parameters vector and Y system output vector. In particular here, we selected as the system output $Y(t) = \frac{dx}{dt}$ to predict the vehicle velocity with respect to the x axis and the measurement matrix consisted of a subset of the measures in M_i defined as $M = [v_m(t)x(t)y(t)\phi(t)]$. Since this model considers 4

[1]This delay is associated to car dynamics as well as internal controller that operates in the system when velocity profiles are set as reference

measures, the system can be described in terms of a Multi input multi output (MIMO) transfer function matrix

$$Y(s) = \begin{bmatrix} h_1(s) & h_2(s) & h_3(s) & h_4(s) \end{bmatrix} \begin{bmatrix} v_m \\ x \\ y \\ \phi \end{bmatrix} \quad (7)$$

where the associated input output transfer functions estimated are

$$h_1(s) = \frac{-4.122s^2 + 49.61s - 4.098}{s^2 + 6.211s + 26.79}$$

$$h_2(s) = \frac{20.55s^2 - 110.4s + 12.08}{s^2 + 37.03s + 111.7}$$

$$h_3(s) = \frac{-0.06946s^2 - 0.06244s + 0.0002256}{s^2 + 1.416s + 0.5473}$$

$$h_3(s) = \frac{-22.3s^2 + 371.5s - 26.08}{s^2 + 6.59s + 18.78}$$

Model presented in 7 had an associated fit percentage of 85.67%

E. Validation

1) Internal Controller Validation: After building the model and validating the model on a test set, we compare our model with the model currently implemented in the car. The current model switches between a kinematic model and a dynamic model depending on the state of the vehicle. The kinematic model is given by equation 6 and has inputs $u = [v, \delta]$ with $z = [x, y, \theta]$. L is a parameter denoting the wheel base of the vehicle.

$$\dot{x} = \begin{bmatrix} v \sin\theta \\ v \cos\theta \\ \dfrac{v \tan\delta}{L} \end{bmatrix} \quad (8)$$

The commanded velocity and steering angles as measured in the experiments are used to determine the outputs of the kinematic model. We take the derivative of the position vectors using the Pythagorean Theorem to get the velocity from the model. This velocity is compared with the velocity we obtain using the transfer function computed above and an error is calculated (see results in figures 4 and 5). We display the results below for comparison between the new model and the kinematic model for one experiment:

Model Type	Mean Square Error
Kinematic Model	1.0972
Transfer Function	0.7430

Therefore, our model improved upon the accuracy given by the existing kinematic model.

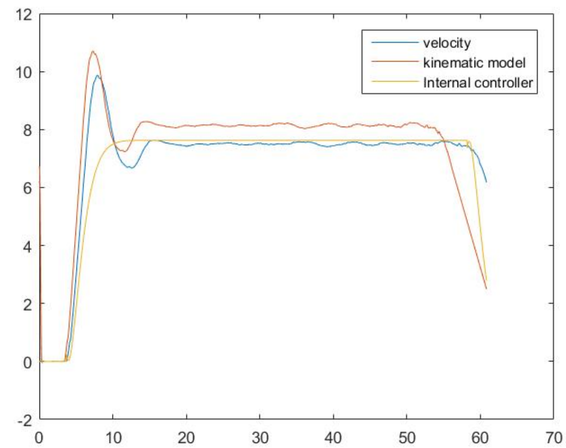

Fig. 4. A plot of the measured velocity with the velocity as computed from the kinematic model and the transfer function model of the internal controller of the car

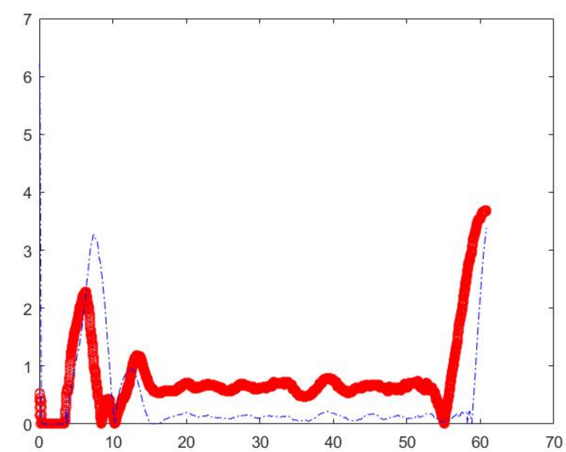

Fig. 5. A plot of the absolute value of the error between the measured velocity and both the velocity computed from the kinematic model and the velocity computed from the transfer function model

2) Car Dynamics Model Validation: We also compare the outputs of the new car dynamics model with the results from the current model in the car. We use the results from one experiment to obtain a position derivative in the x direction. The steering angle in radians is computed and is inputted alongside the commanded velocity into the kinematic model. This vector is compared with the output from the new model to obtain the following results in one experiment:

Model Type	Mean Square Error
Kinematic Model	79.8702
Transfer Function	13.1558

Much of the error in the transfer function was due to noisy signals at the beginning (as can be seen in figure 6). Overall, the new model is more accurate than the previously

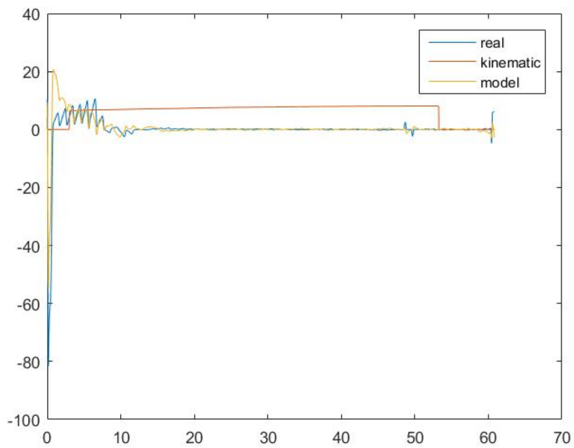

Fig. 6. A plot of the measured x position derivative, the x position derivative as computed by the kinematic model, and the x position derivative as computed by the transfer function model developed above

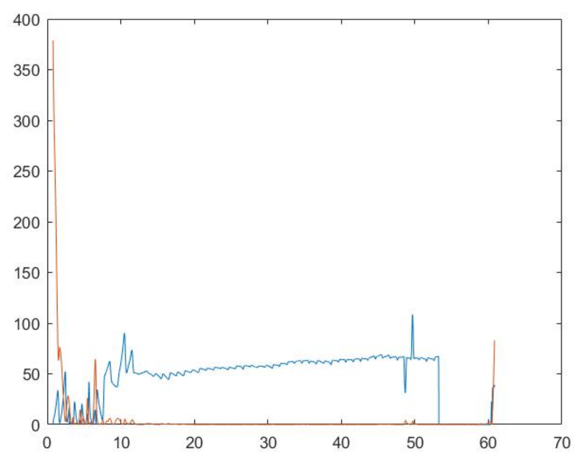

Fig. 7. A plot of the absolute value of the error between the measured dx the dx computed from the kinematic model and the new transfer function model developed in the previous section

implemented kinematic model (errors can be seen in figure 7).

III. Conclusions and Future Works

The transfer function used to represent the system is more accurate and more computationally efficient than the previous kinematic model. Further research in the development of the mathematical model could involved further characterization of the error inherent in the CAT Vehicle's sensors, which would improve the accuracy of the model. Another potential area for future work could involve more exhaustive system identification of the vehicle, in which the experimental data gathered would test a wider range of typical driving conditions.

Acknowledgments

This research is supported by the National Science Foundation and the Air Force Office of Scientific Research, under awards IIS-1262960 and CNS-1253334. Research conducted through the University of Arizona's CAT Vehicle REU program.

References

[1] Mir Aamir Abbas, Ruth Milman, and J Mikael Eklund. Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pages 1–6. IEEE, 2014.

[2] Sterling J Anderson, Steven C Peters, Tom E Pilutti, and Karl Iagnemma. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International Journal of Vehicle Autonomous Systems*, 8(2-4):190–216, 2010.

[3] Panos J Antsaklis, James A Stiver, and Michael Lemmon. Hybrid system modeling and autonomous control systems. In *Hybrid Systems*, pages 366–392. Springer, 1993.

[4] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous systems*, 32(1):1–16, 2000.

[5] David M Bevly, Jihan Ryu, and J Christian Gerdes. Integrating ins sensors with gps measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):483–493, 2006.

[6] Yen-Lin Chen, Chuan-Tsai Lin, Chung-Jui Fan, Chih-Ming Hsieh, and Bing-Fei Wu. Vision-based nighttime vehicle detection and range estimation for driver assistance. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 2988–2993. IEEE, 2008.

[7] Paolo Falcone, Francesco Borrelli, H Eric Tseng, Jahan Asgari, and Davor Hrovat. A hierarchical model predictive control framework for autonomous ground vehicles. In *2008 American Control Conference*, pages 3719–3724. IEEE, 2008.

[8] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.

[9] Cheng-Yang Fu, Chun-Kun Wang, and Eunbyung Park. A survey of computer vision research for automotive systems. 2015.

[10] Alison Gray, Yiqi Gao, Tao Lin, J Karl Hedrick, H Eric Tseng, and Francesco Borrelli. Predictive control for agile semi-autonomous ground vehicles using motion primitives. In *American Control Conference (ACC), 2012*, pages 4239–4244. IEEE, 2012.

[11] Davor Hrovat, Stefano Di Cairano, H Eric Tseng, and Ilya V Kolmanovsky. The development of model predictive control in automotive industry: A survey. In *Control Applications (CCA), 2012 IEEE International Conference on*, pages 295–302. IEEE, 2012.

[12] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 163–168. IEEE, 2011.

[13] Lennart Ljung. System identification: theory for the user. *Englewood Cliffs*, 1987.

[14] Jihan Ryu and J Christian Gerdes. Integrating inertial sensors with global positioning system (gps) for vehicle dynamics control. *Journal of Dynamic Systems, Measurement, and Control*, 126(2):243–254, 2004.

[15] Dirk E Smith and John M Starkey. Effects of model complexity on the performance of automated vehicle steering controllers: model development, validation and comparison. *Vehicle System Dynamics*, 24(2):163–181, 1995.

[16] Yongsoon Yoon, Jongho Shin, H Jin Kim, Yongwoon Park, and Shankar Sastry. Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*, 17(7):741–750, 2009.

[17] Melanie N Zeilinger, Davide M Raimondo, Alexander Domahidi, Manfred Morari, and Colin N Jones. On real-time robust model predictive control. *Automatica*, 50(3):683–694, 2014.

[18] Kun Zhang, Jonathan Sprinkle, and Ricardo G Sanfelice. Computationally aware control of autonomous vehicles: a hybrid model predictive control approach. *Autonomous Robots*, 39(4):503–517, 2015.

[19] Kai Zheng, Yu Zheng, Xing Xie, and Xiaofang Zhou. Reducing uncertainty of low-sampling-rate trajectories. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1144–1155. IEEE, 2012.