Recovering Models of a Four-Wheel Vehicle Using Vehicular System Data

Jonathan Sprinkle
J. Mikael Eklund
Humberto Gonzalez
Esten Grotli
Pannag Sanketi
Michael Moser

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-92
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-92.html

August 12, 2008

Acknowledgement

This work was supported in part by the Center for Hybrid and Embedded Software Systems (CHESS)at UC Berkeley, which receives support from the National Science Foundation (NSF awards #CCR-0225610 (ITR),#0720882 (CSR-EHS: PRET), #0647591 (CSR-SGER), and 0720841 (CSR-CPS)), the U. S. Army Research Office (ARO #W911NF-07-2-0019), the U. S. Air Force Office of Scientific Research (MURI #FA9550-06-0312 and AF-TRUST #FA9550-06-1-0244), the Air Force Research Lab (AFRL), the State of California Micro Program, and the following companies: Agilent, Bosch, DGIST, Lockheed Martin, National Instruments, and Toyota.

Recovering Models of a Four-Wheel Vehicle Using Vehicular System Data

Jonathan Sprinkle, J. Mikael Eklund, Humberto Gonzalez, Esten Grøtli, Pannag Sanketi, Michael Moser, S. Shankar Sastry

Abstract— **This paper discusses efforts to parameterize the actuation models of a four-wheel automobile for the purposes of closed-loop control. As a novelty, the authors used the equipment already available or in use by the vehicle, rather than expensive equipment used solely for the purpose of system identification. After rudimentary measurements were taken of wheelbase, axle width, etc., the vehicle was driven and data were captured using a controller area network (CAN) interface. Based on this captured data, we were able to estimate the feasibility of certain closed-loop controllers, and the models they assumed (i.e., linear, or nonlinear) for control. Examples were acceleration and steering. This work served to inform the separation of differences in simulation and vehicle behavior during vehicle testing.**

I. INTRODUCTION

A major complexity of vehicle control is an accurate model of the vehicle for controller design, and simulation. Poor vehicle models can result in unstable behavior when applied to hardware, resulting in unsafe situations for those involved, or frustrating demonstrations that diverge significantly from simulation results. As a result, significant hardware-in-the-loop testing is necessary prior to control system design, to minimize the risk.

This is undesirable from many aspects. First, hardware-in-the-loop experiments are costly in terms of personnel time and equipment, as well as any facilities which must be rented. Additionally, it predicates the implementation of algorithms for higher-level performance on already known or well-understood platforms upon which those algorithms will run. Finally, it can encourage *ad hoc* design of controller software that "seems" to be accurate, when in fact it is not. This latter, especially, can be disastrous for a project if the accuracy of the controller is not appropriately similar in hardware and in simulation, and a significant amount of algorithm design has been performed in simulation. All engineering exercises, such as controller tuning, parameter identification for lookup tables, etc., is in question.

In this paper, we address each of these concerns by describing our technique for recovering the model of a four-wheel drive vehicle. Through this use, we were able to provide models describing the estimated behavior in various operational modes of the vehicle (acceleration, steady driving, slow-speed driving), and perform some assessment of how closely fit the model is to actual performance. The advantage to our approach is that we gathered data directly from the vehicle's CAN bus, and from various external localization sources (primarily GPS and inertial data). We present the graphs of our estimates, and actual performance, under various circumstances.

A. Significance of the problem

This work was motivated by the experience of our research team, who successfully created a significant body of high-level algorithms, each of which utilized a common low-fidelity model for steering and velocity control that interfaced with open-source vehicle simulators. As hardware-in-the-loop testing continued, it became apparent that many of the algorithms, while running in simulation, were unable to terminate, or produced significant functional errors, due to the fact that they were tested and tuned using a simulator that did not match the vehicle platform. While simulation had enabled a "bootstrap" solution to tuning the parameters, significant additional tuning (with the hardware-in-the-loop) was required, which consumed enormous resources both in time and personnel.

At runtime, vehicle operators were faced with the following quandary: was the previous failure due to a logical error, optimization error, open-loop control error, or closed-loop control error? Further, assumptions made in high-level controllers on maximum turn rate, acceleration, etc., may not reflect the vehicle's actual behavior. Without confidence in the model used by closed-loop controllers, significant deviation could take place in vehicle trajectory (especially at low-speeds), which is not noticed by a human driver watching the road, but is compounded by a predictive controller that may be estimating potential position in 10-15 seconds. In order

This work was supported in part by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley, which receives support from the National Science Foundation (NSF awards #CCR-0225610 (ITR),#0720882 (CSR-EHS: PRET), #0647591 (CSR-SGER), and #0720841 (CSR-CPS)), the U. S. Army Research Office (ARO #W911NF-07-2-0019), the U. S. Air Force Office of Scientific Research (MURI #FA9550-06-0312 and AF-TRUST #FA9550-06-1-0244), the Air Force Research Lab (AFRL), the State of California Micro Program, and the following companies: Agilent, Bosch, DGIST, Lockheed Martin, National Instruments, and Toyota. The Sydney-Berkeley Driving Team was supported in part by Rio Tinto, Komatsu, Toyota, ZeroC, and Advantech. Additional in-kind support was provided in the form of discounts on equipment from the following manufacturers: SICK, NovAtel, and Honeywell.

J. Sprinkle is with the University of Arizona, Tucson, AZ 85721-0104, USA sprinkle@ECE.Arizona.Edu

J.M. Eklund is with the Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, Canada Mikael.Eklund@uoit.ca

H. Gonzalez, P. Sanketi, and S.S. Sastry are with the University of California, Berkeley, CA {hgonzale, sastry}@EECS.Berkeley.Edu, pannag@ME.Berkeley.Edu

E. Grøtli is with the University of Trondheim, Norway Esten.Grotli@itk.ntnu.no

Michael Moser is with the Australian Centre for Field Robotics, University of Sydney, Sydney Australia m.moser@acfr.usyd.edu.au

to place high confidence in the vehicle model used by the closed-loop and open-loop controllers[1], there must be some focus on determining the model for actuating the vehicle in question.

B. Novelty of solution

As the reader will doubtless suggest, this work seems to duplicate results achieved in the field of system identification [1] [2], and there exist plentiful patents for performing work similar to this using state-of-the-art sensors [3]. What makes this work unique is its dependency on information gathered directly from the vehicle's data stream, and other relatively-unsophisticated sensors that were in use for high-level autonomy. We perceive a difficulty for many researchers who might need to decide whether their models are sufficiently approximate for their purposes, but who do not have budget or time to produce advanced system identification results through specialized equipment, even though those results would be highly accurate.

Another important reason to discover sufficiently approximate models (rather than highly accurate models) is to allow predictive controllers to utilize models that are rapid to compute, and accurate for time horizons on the order of 10 seconds.

The low-cost and low-effort solution presented here consists of analyzing data readily available through the vehicle's CAN bus, and through the GPS and inertial systems on board the vehicle. In fact, researchers in autonomous vehicles may already have these data available from years of research, but not have analyzed it yet.

Although these GPS and inertial systems are not inexpensive, they are considered a standard part of any autonomous vehicle's package, and we therefore consider them to be readily available. For systems where only dead-reckoning is used, we do not require that the information be available in real-time, so *a posteriori* results for localization, such as SLAM, are certainly sufficient to produce the necessary datasets.

II. SCOPE

This section describes the models we expected to find, and the equations governing certain models that were assumed, but of which we did not know the parameters. In addition, the variances between control inputs and observables is mentioned.

A. Steering Model Formalization

The Ackermann vehicle model [4] provides an approximation of vehicle motion for vehicles with four tires, and steering control for the front two tires. Although the model includes for different angles for the tires, it does not account for mechanical vehicle stability and wear items, such as the transmission slip, and limited slip differential, when accounting for motion.

Generally, though, the Ackermann Model is a good first-order approximation to vehicle motion. In fact, for control inputs, and predictive control, it is sufficient to consider an even simpler model—a bicycle model—where the left and right tires are combined into a "virtual" bicycle, with one front and one rear tire at the center of the vehicle's longitudinal axis. Figure 1 shows the Ackermann Model, and bicycle model. The motion equations of a bicycle model are presented in (1) where (x,y) are the cartesian coordinates of the front wheel, v is the speed of the body of the bicycle, ψ is the angle of the body of the bicycle with respect to a fixed frame, δ is the angle of the front wheel with respect to the body of the bicycle and b is the length between front and rear wheels.

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\left(\psi(t)+\delta(t)\right) \\ \dot{y}(t) &= v(t)\sin\left(\psi(t)+\delta(t)\right) \\ \dot{\psi}(t) &= \frac{1}{b}v(t)\sin\left(\delta(t)\right) \end{aligned} \tag{1}$$

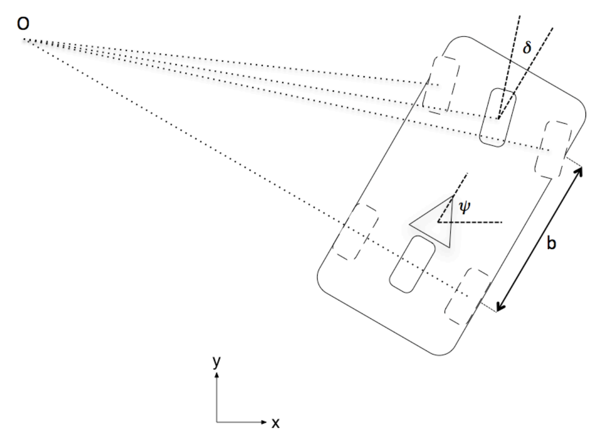

Fig. 1. The Ackermann Model, with a simplified bicycle model, used for control, and predictive modeling of the vehicle's trajectory of motion.

B. Steering Wheel versus Wheel Angle

The two steered tires in a four-wheeled vehicle are not always turned at the same angle. Although simulation and actual behavior of the vehicle reflects this difference, the control input (from the steering wheel) provides only on value. In addition to minor discrepancies introduced in the rotational motion of the steering wheel to the road tires, trigonometric-related nonlinearities are introduced by the rack and pinion. Thus, the relationship between steering wheel angle and the road tires is not easily calculated by testing the limits of the vehicle's steering, and performing careful measurements of the tire angles from the longitudinal direction of travel.

In this paper, we use δ_t as the angle of the tire (relative to the vehicle's heading angle, ψ), and $\delta_{sw} \in [\delta_{sw_{min}}, \delta_{sw_{max}}]$ as the angle of the steering wheel. For most four-wheel vehicle, $\delta_{sw_{max}} - \delta_{sw_{min}} \approx 5\pi$

[1]It is important to note that while the use of the model is important, the open-loop and closed-loop controllers may use different fidelity models for their purposes. Generally, though, parameters for the low-fidelity models are based on the linearization or approximation of the high-fidelity model. Our work is different in this respect, by gathering the low-fidelity model from data.

C. Acceleration Models

The accelerator models are important for controlling speed in a closed-loop manner. Consider an accelerator model where the accelerator input is defined as $u_a \in [0, N]$, where $N \in \mathbb{Z}$.

D. Small-Velocity Models

Because modern fuel-injected vehicles have an idle speed which can drive the vehicle forward if the brake is not depressed, velocities on the order of 2 m/s are difficult to achieve. In such scenarios, chattering between the accelerator and brake is common. Estimating the behavior of a velocity controller at slow speeds can be very important, to reduce this chattering.

E. Data Availability

The data were obtained from 3 sources: the vehicle's CAN bus, a *NovAtel SPAN* system and the custom actuation system developed by the Sydney-Berkeley Driving Team. The vehicle's CAN bus provided measurements of the internal state of the car such as both pedal positions (accelerator and brake), the independent speed of each wheel, the revolutions per minute of the engine and the gear used at each time step. The *NovAtel SPAN* system provided accurate measurements for position, velocities (linear and angular) and accelerations (linear and angular). The actuation system was used in a tele-operation mode, eliminating human errors during the data acquisition process and allowing a fast and continuous sampling of the steering wheel angle and the signal applied to both pedals.

The time synchronization of all the sources was performed by a computer running the *QNX* real-time operating system.

III. EXPERIMENTS PERFORMED

In order to gather useful data, we put the vehicle through several motion plans, including several regular-use scenarios (i.e., road driving by a human).

A. Steering Model Experiments

The following experiments were performed to test the steering model approximation.

1) Driving in an arc: The steering controller was given a series of reference inputs for δ_{sw}, while driving in a clockwise arc. The reference inputs were provided while driving at a longitudinal velocity of 2.8 m/s. After each reference input, the reference was set back to 0 (i.e., straight ahead). This experiment was repeated while traveling in a counter-clockwise arc at 2.4 m/s. The pair of experiments was repeated at 5.2 m/s.

2) Driving in a circle: The reference value for δ_{sw} was gradually increased to turn clockwise until full lock was achieved (i.e., the tightest turn possible). The vehicle held this trajectory until a circle was traversed. This experiment was repeated for a counter-clockwise circle.

3) Figure Eight Turns: Human driver, performing all steering, acceleration and braking inputs. The vehicle was put through figure eight turns, at a relatively constant velocity. Only "Low" gear was used.

B. Acceleration/Speed Model Experiments

The following experiments were performed to exercise the acceleration and speed models.

1) Manual Steering, autonomous speed (single ramps): Driving on an almost straight road, accelerator and brake in autonomous mode, steering wheel in manual mode. The speed reference was a ramp from 0 m/s to 6 m/s, maintain for a short time and then another ramp from 6 m/s to 0 m/s. This experiment was duplicated with ramps of $[0,4]$ m/s, and $[0,8]$ m/s.

2) Manual Steering, autonomous speed (sequence of ramps): Similar to Section III-B.1, but with a sequence of ramps as $[0,4]$ m/s, $[4,8]$ m/s, $[8,4]$ m/s, and $[4,0]$ m/s.

3) Open-loop response of throttle and brake: This experiment was performed without the feedback loop of the low-level speed controller. The vehicle was accelerated, braked to a hale, accelerated, and braked to a halt. This experiment was repeated.

IV. STEERING MODEL RECOVERY

To achieve a simple model of the steering model, we determine the tire angle from calculating the difference in tire speed between the front tires, left and right. This information is captured from the CAN, as described in Section II-E.

A. Model Approximation

The estimate is based on the following model:

$$v_l = r_l \omega \tag{2}$$

$$v_r = r_r \omega \tag{3}$$

$$r_r = r_l + C_{lr} \tag{4}$$

$$\omega = \frac{v_r - v_l}{C_{lr}} \tag{5}$$

$$\delta_t = \frac{\omega}{v} \tag{6}$$

where v_l and v_r the respective velocities of the left and right tires, r_l and r_r are the radii of the left and right tires of the angle of the vehicle's turn, C_{lr} is the centerline distance between the two front tires, ω is the rotational velocity of the logical middle tire, and δ_t is the steering angle of the logical middle tire.

B. Parameter Identification

In using these models, we can derive parameters for the following approximate model for the tire angle:

$$\delta_{t_{\text{estimate}}} = \Delta_{\text{offset}} + \Delta_{\text{scale}} \delta_{sw} \tag{7}$$

Where $\delta_{t_{\text{estimate}}}$ is an estimate of the logical center tire's angle, Δ_{offset} and Δ_{scale} are constants representing the linear offset and scaling factor (respectively), and δ_{sw} is the measured steering wheel angle. This methodology is described in greater detail in [5] and is based on foundational work found in [6].

To derive the parameters, we use offline optimization routines to compare the actual value of the vehicle's position, with our estimate of the vehicle's position using this

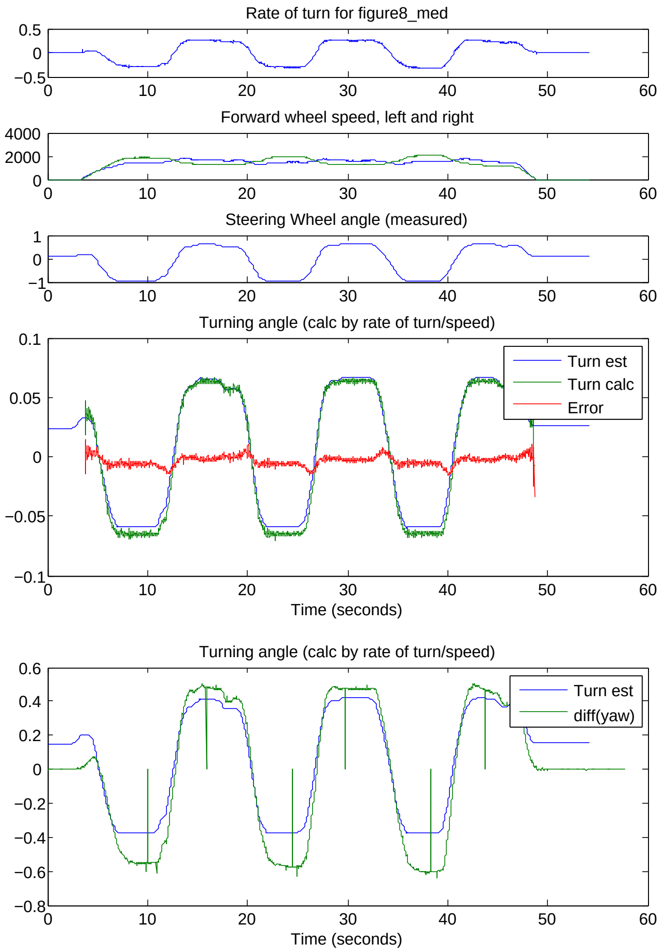

Fig. 2. A model of the car turn angle, based on steering wheel angle measurements. The top graph shows the rate of turn (calculated in (5)). The next graph displays the forward wheel speeds. Note here that the nonlinearities in the rack and pinion are evident at the most significant steering wheel angle, even though the wheel is not fully locked. The next graph shows the measured steering wheel angle, based on CAN data. The final two graphs compare the estimated value with two calculated turn angles: that of the values in the top graph divided by the speed, and the bottom graph reflecting yaw rates gathered from the differential GPS unit.

simplified steering model for steering wheel input control. Measurement errors are minimized by selecting as optimization inputs the Δ_{offset} and Δ_{scale} parameters, to minimize the difference of the actual measured tire angle and $\delta_{t_{\text{estimate}}}$ over the entire dataset.

C. Results of Steering Model Recovery

Figure 2 shows the results of processing data from experiments described in Section III-A, specifically Experiment III-A.3. The sinusoidal graphs are indicative of the figure eights as they were traversed. Most interesting are the bottom two graphs in the figure, which show the estimated turning angle ($\delta_{t_{\text{estimate}}}$) compared against turning angles calculated from two sources: the calculated tire angle δ_t (see (6)), and based on differential yaw measurements.

Note that the error for comparison against the calculated tire angle (based on other measurements) is proportional to the rate of change of turn. The second calculation, however, is susceptible to noise from the inertial system (note especially the large spikes). The estimate does not differ from the calculated measurement by more than 5%. The calculated values for our particular vehicle were $\Delta_{\text{offset}} = 0.017$, and $\Delta_{\text{scale}} = 0.08$.

V. ACCELERATOR MODEL RECOVERY

We consider several basic zones of execution, which were chosen based on our platform-specific knowledge of vehicle performance, and are not necessarily of interest to every kind of four-wheeled vehicle, nor are they optimal for this vehicle. However, they were indicated as discrete modes of execution based on user experience. These zones were the idle-motion $v \in [0,2]$ m/s, low-velocity $v \in (2,5]$ m/s, and high-speed $v \in (5,20]$ m/s.

As introduced in Section II-C most fuel-injected vehicles will move at a constant speed at idle (over level-ground). Thus, to maintain a velocity $v_0 \in (0, v_{\text{idle}}]$ requires actuation of the brake. In our case, we presume that $v_{\text{idle}} \approx 2$ m/s.

The low-velocity group is chosen specifically for its upper-bound. In our case, the velocity of 5 m/s (approximately 11.2 miles/hour) is near the our desired top speed for “Low” gear in the transmission. Thus, with actuator control for gear shifting, we could conceivably shift between controller models when shifting into “Middle” gear.

The actual vehicle model for velocity (with the accelerator control input) is highly nonlinear due to vehicle inertia, engine operating temperature, transmission gear, differential slip, tire traction, and many other factors. However, using PID control to maintain velocity (above the idle velocity) is actually somewhat trivial. The difficulty, therefore, is providing an approximate model, suitable for the high-performance requirements of predictive controllers over a finite horizon, while not significantly compromising accuracy over that same horizon. Without this suitable approximations, predictive controllers could choose suboptimal paths in early timesteps, resulting in a cul-de-sac trajectory choice.

A. Model approximation

Given the high nonlinearities expected, we treat the accelerator model as a black box, and evaluate it over our arbitrary zones of speed. Nonetheless, we provide the following definition for optimization:

$$a_{\text{estimate}} = A_{\text{offset}} + A_{\text{scale}} a_{\text{p}} \tag{8}$$

where a_{estimate} is the accelerator estimate, A_{offset} and A_{scale} are constants representing the linear offset and scaling factor (respectively), and a_{p} represents the measured accelerator angle.

B. Parameter Identification

To identify parameters over these ranges, we collect data over a wide range of system exercise (described in Section III), and perform offline optimization of (8) with optimization inputs of A_{offset} and A_{scale}, to minimize the difference between actual measured, and a_{estimate}.

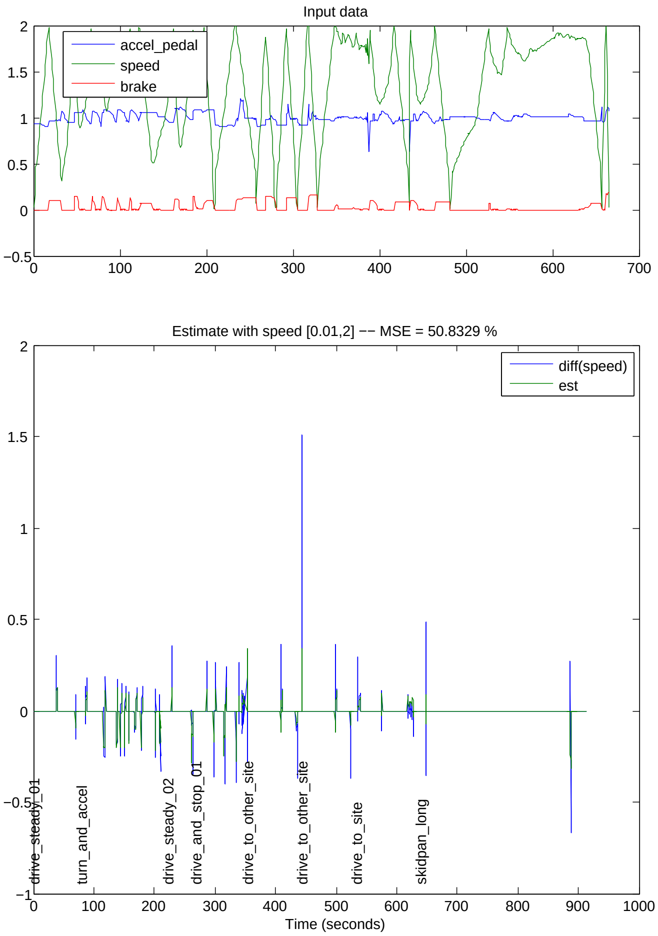

Fig. 3. A model of the car speed/acceleration at very slow speed, between $(0, 2]$ m/s. The upper graph shows input data of the accelerator pedal, speed, and brake setting. In the lower graph, estimates of the acceleration are compared with the measured speed from the differential GPS unit. The estimate of the acceleration is based on the estimated accelerator angle.

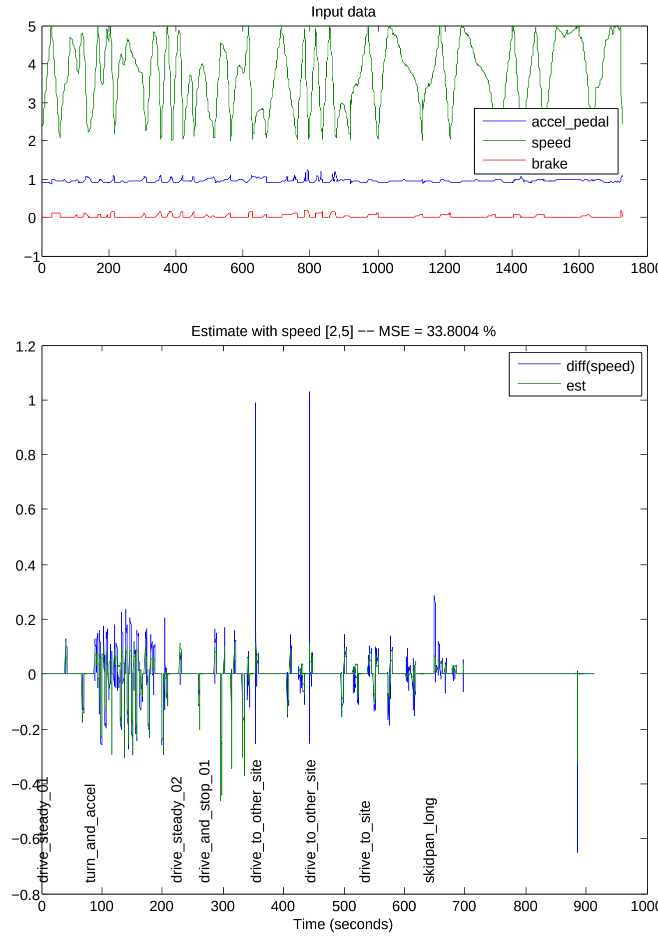

Fig. 4. A model of the car speed/acceleration at moderate speed, between $(2, 5]$ m/s. The upper graph shows input data of the accelerator pedal, speed, and brake setting. In the lower graph, estimates of the speed are compared with the measured speed from the differential GPS unit. The estimate of the acceleration is based on the estimated accelerator angle.

C. *Results of Accelerator Model Recovery*

The recovery of accelerator performance is significantly less accurate, in terms of percentage, than that of the steering wheel relationship to the turning angle. This is due in part to the relative complexity of each problem (there is a near mechanical linkage between the steering wheel and tires, whereas the drivetrain is separated from the accelerator pedal by many nonlinear devices).

Figures 3, 4, and 5 show the results of many collected datasets that were selected from experiments described in Section III-B. The data are extracted based on the speed, such that all datasets would be reconstructed if data were re-assimilated from the three graphs (i.e., subsets of each range of the zones described previously were removed). Since our model makes no assumptions on continuity or inertia, this poses no threat to testing our model of the accelerator.

Figure 3 shows acceleration estimates, based on estimated accelerator pedal values as calculated in (8), for velocities between 0 and 2 m/s. The mean-square error is 50.8329%, based on calculated speed compared to that of the differential GPS speed, as recorded.

Likewise, Figure 4 shows estimated and actual accelerations subset of speeds between 2 m/s and 5 m/s. In this case, the MSE is much lower, 33.8004%, based on the same comparisons. Finally, Figure 5 displays a much higher MSE, at 71.0085%. The higher variance of these acceleration estimates, and how performing this analysis informed the vehicle's controllers and models, is addressed in the next section.

VI. ANALYSIS

As would be expected, based on the simple models we were deriving from these datasets, the models for these complex behaviors are not of sufficient fidelity for simulation. What we have calculated is a simple relationship between control inputs for our vehicle's simulated behavior: steering wheel angle to manipulate the turn angle, and accelerator pedal to manipulate the velocity.

Since the closed loop controllers that would most likely implement such behaviors for an autonomous system would likely be linear themselves, we can now surmise two important points: (a) the steering wheel relationship to turn angle

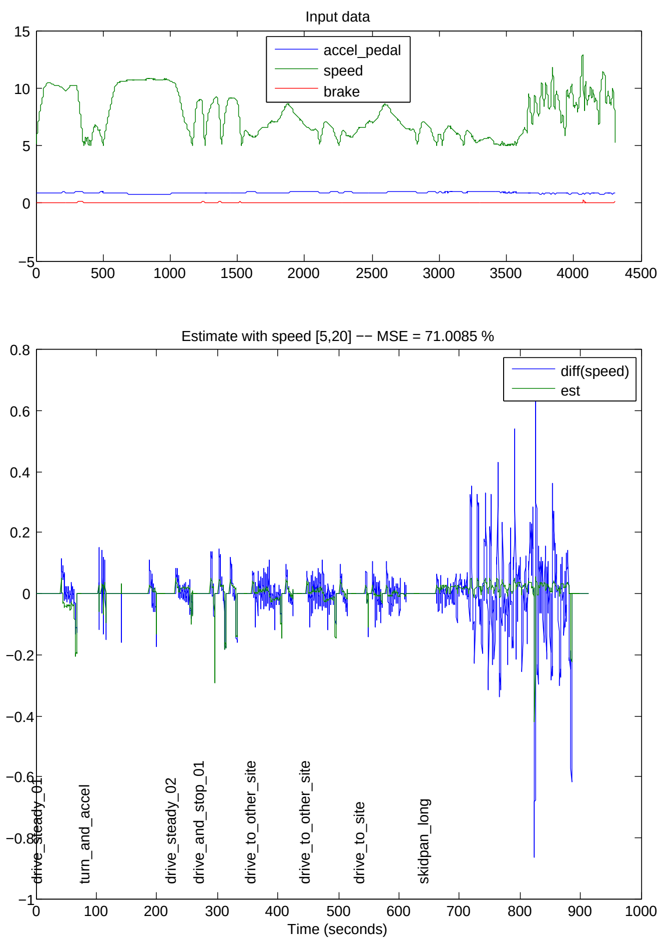

Fig. 5. A model of the car speed/acceleration at relatively high speeds, between $(5,20]$ m/s. The upper graph shows input data of the accelerator pedal, speed, and brake setting. In the lower graph, estimates of the acceleration are compared with the measured speed differential from the differential GPS unit. The estimate of the acceleration is based on the estimated accelerator angle.

is appropriate for such linear controllers, and (b) control of the speed by the simple accelerator model is sufficient, but will likely experience some chattering and lag to achieve the desired velocity.

Raising these two points is extremely important for controllers who perform online optimization. In fact, it helps system designers understand that controlling the steering is much easier than controlling velocity, and as such velocity control should have a higher cost.

Further, it is clear from the MSE of the three zones of vehicle speed, that the simple accelerator model is the most accurate between $(2,5]$ m/s. As such, a vehicle controller should attempt to remain in this zone as much as possible.

Two immediate ways to enforce this speed band are to set a minimum speed for the vehicle's velocity, which is overridden in case of dangerous obstacles or commands to stop, or to consider the speed model as a hybrid controller, where transitions to the lower or higher speeds are permitted, but only when the horizon indicates that the acceleration changes are not expected.

Finally, due to the knowledge that very low speeds near idle are prone to errors and lags in setting velocities, vehicle designers will be less-likely to try to test at slow speeds for controller tuning, since it should be apparent that such tunings will be inaccurate for higher speeds. Without this knowledge, the controller implementation team could spend tremendous hours scratching their heads as to why turning up the speed works in simulation, but not in hardware.

VII. CONCLUSIONS

We have presented motivation, and justification, for developing an oversimplified model for vehicle acceleration and turning angle estimates, based on measurable values (accelerator pedal, and steering wheel). We showed that such models are certainly not high-fidelity in terms of accuracy, but that since closed loop controllers may be linear, that understanding when or why there are discrepancies can inform high-level modelers of additional costs to consider in their optimizations.

To perform these calculations and analyses, we discussed how using existing data from the vehicle, such as CAN bus data and GPS measurements, the data can be captured without the need for significant equipment outlay. Further, since many high-fidelity system identification models will not run in real-time, we have provided a rough approximation for these closed-loop control input relationships that could be used in predictive controllers to assign a real-time cost based on the current state, and known errors in estimation.

VIII. ACKNOWLEDGEMENTS

This work was due to the tremendous effort of the Sydney-Berkeley Driving Team, which entered the DARPA Urban Challenge in 2006. Special thanks for that effort are deserved for Dr. Ben Upcroft, whose contributions provided the ability to gather much of the data in this paper. The Sydney-Berkeley Driving Team was supported in part by Rio Tinto, Komatsu, Chess at UC Berkeley, Toyota, ZeroC, and Advantech. Additional in-kind support was provided in the form of discounts on equipment from the following manufacturers: SICK, NovAtel, and Honeywell.

REFERENCES

[1] K. Yi and K. Hedrick, "Observer-based identification of nonlinear system parameters," *Journal of Dynamic Systems, Measurement, and Contol*, vol. 117, no. 2, pp. 175–182, 1995.

[2] A. B. Proca and A. Keyhani, "Identification of power steering system dynamic models," *Mechatronics*, vol. 8, no. 3, pp. 255–270, April 1998.

[3] T. Kawabe and K. Ito, "Model solving type vehicle steering control system with parameter identification," U.S. Patent, January 1988, Nissan Motor Co., Ltd.

[4] E. Nebot, "Navigation system design," Lecture Notes, May 2005, center of Excellence for Autonomous Systems, University of Sidney, Australia.

[5] J. M. Eklund, M. Korenberg, and P. McLellan, "Nonlinear system identification and control of chemical processes using fast orthogonal search," *Journal of Process Control*, vol. 17, no. 9, pp. 742–754, October 2007.

[6] M. J. Korenberg, "A robust orthogonal algorithm for system identification and time-series analysis," *Biol. Cybernetics*, vol. 60, pp. 267–276, 1989.