A Fast and Reliable Reduced Order Model for Large Scale Vehicle Simulation

Huzaifa Unjhawala[1], Thomas Hansen[1], Luning Fang[1], Zhenhao Zhou[1], Shouvik Chatterjee[1], Jinlong Wu[1], Radu Serban[1], Dan Negrut[1]

[1] Department of Mechanical Engineering,
University of Wisconsin - Madison, Madison, Wisconsin, United States
Email: {unjhawala, thansen8, lfang9, zzhou292, schatterjee9, jinlong.wu, serban, negrut}@wisc.edu

EXTENDED ABSTRACT

1 Introduction

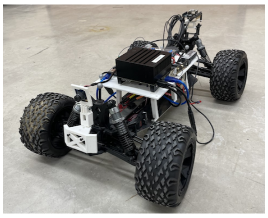
Simulation is useful in engineering system design as it can shorten time-to-market, reduce costs, and provide more comprehensive testing. In simulating complex non-linear dynamic systems such as a vehicle, one usually faces a trade-off between model *accuracy* and *simplicity*. While high fidelity models offer higher accuracy, lower fidelity models are easier to implement and faster to simulate. Despite the burgeoning compute power at our disposal, high-fidelity models are often times too computationally expensive for applications such as model predictive control, state estimation, or large-scale traffic simulation. In this contribution we seek to address the complexity/simulation-speed concern by developing reduced order models (ROMs) that are fast yet sufficiently accurate. We propose a ROM that is fast and accurate for a large number of maneuvers that a vehicle usually performs in road traffic, state estimation, sensitivity analysis, or for control design. Specifically, we establish a ROM whose accuracy is acceptable over a broad range of state and input commands, and whose "training" requires calibration data that is easily available. We use an approach in which we first fix the degree of freedom count of the model. Then, to make this ROM accurate, we identify model parameters (θ) by using a *Bayesian inference* framework [1]. The Bayesian approach views the unknown parameters as random variables and aims at producing the associated conditional probability distributions, i.e., the posterior distribution $\pi(\theta) = p(\theta|y)$, given calibration data y. We use motion capture and thus calibration data is noisy, making Bayesian inference preferable over optimization techniques. To demonstrate the approach, we generate a ROM for an 1/6th scale vehicle, see Fig. 1a.

2 Methodology

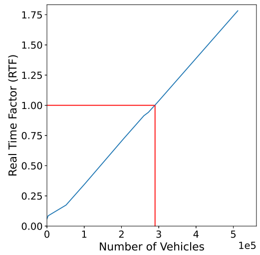
The general problem is cast as $y = \mathscr{G}(\theta) + \varepsilon, \quad \varepsilon \sim N(0, \Gamma)$, where $\mathscr{G} : \mathscr{X} \mapsto \mathscr{Y}$ denotes the reduced order vehicle model with unknown parameters $\theta \in \mathscr{X}$; $y \in \mathscr{Y}$ represents the available data with noise ε that follows a zero-mean normal distribution with a covariance matrix Γ; and $\mathscr{X}$ and $\mathscr{Y}$ denote complete normed vector spaces. The goal is to estimate the posterior distribution $p(\theta|y)$ via Bayesian inference $p(\theta|y) \propto \exp\left(-\frac{1}{2}\|y - \mathscr{G}(\theta)\|_{\Gamma}^{2}\right) p(\theta)$, where $p(\theta)$ is the prior distribution that captures existing approximate knowledge (if any) about the unknown parameters θ. We use Sequential Monte Carlo (SMC) [2] to draw samples from the posterior distribution $p(\theta|y)$. Compared to the traditional Markov Chain Monte Carlo (MCMC) approach, SMC displays better sampling of multi-modal distributions, and is more computational efficient. The ROM ($\mathscr{G}$) is a four-wheel vehicle model [3] augmented with non-linear TMeasy tires [4], a map-based engine model and a kinematic powertrain model. The ROM has lateral, longitudinal, yaw and roll DOF's. Additionally, each of the TMeasy tires has first order differential equations for the longitudinal and lateral tire deflections along with a differential equation associated with its rotation. We use the relatively complex TMeasy tire model mainly due to two reasons, (i) TMeasy is valid in all driving situations and provides smooth transition from standstill; and (ii) TMeasy tire parameters can easily be deduced by knowing its size, payload and friction coefficient with the road which gives us great priors for parameter identification. The model is written in C++ and exposed to Python via *SWIG*. When benchmarked on an Intel(R) Core(TM) i7-4770K CPU, the model, which is solved using a half-implicit integrator [5] with a step size of 1e-3 (s), is found to be approximately 1000 times faster than real time.

3 Preliminary Results

To demonstrate the ROM-generation approach, we used our Autonomy Research Testbed (ART) vehicle, see Fig. 1a; it is controlled by a Jetson Xavier and uses a 1300kV brushless motor [6]. To produce calibration data, we used an OptiTrack Motion Capture room, where the vehicle's movement could be traced precisely by 13 cameras updating at 100Hz. Here, we calibrate the motor torque and loss curves using two simple maneuvers, namely a straight line acceleration with a ramp and step throttle input, respectively. We apply the same inputs to both the ART and ROM vehicles and use the longitudinal velocity (u) as our data (y). Along with the motor curves, we also sample the standard deviation of the noise (σ_u) in u. We use four SMC chains in parallel to sample 1000 draws of each of our parameters. We assume uniform priors based on knowledge of the motor and ESC specifications over a suitable range for the points in our curves and a half-normal prior over σ_u. The *posterior*, plotted after performing Kernel Density Estimation (KDE) and normalizing the Y-axis, can be seen in Fig. 2a. We evaluate the convergence of the posteriors using diagnostics such as split-$\hat{R}$ and ensure that they are all less then 1.01 [7]. By taking the empirical mean of the posterior distribution of our parameters, we can get the mean posterior motor torque and losses curve (Fig. 2b). We can then sample the posterior distributions to evaluate the model response (see Fig. 2c). We quantify this fit of prediction using the mean of the Root-Mean-Squared-Error (mean-RMSE) between the 100 response lines and the noisy data. The mean of the



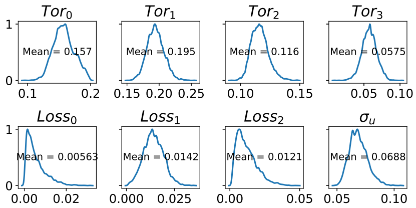
(a) Vehicle run in the Motion Capture room to collect data.



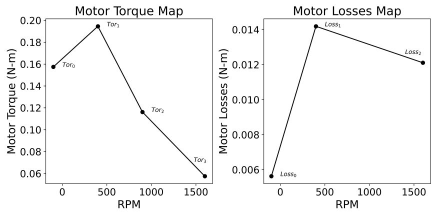
(b) Scaling analysis on the NVIDIA A100 GPU.

Figure 1: Vehicle used for calibration; almost 300,000 ROMs can be run in real time.

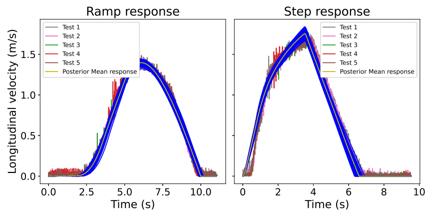
mean-RMSE over 5 test runs was found to be 0.014 over the ramp response and 0.020 for the step response. The calibration of other parameters, especially those that play a role in lateral dynamics is ongoing. We extended the ROM to the GPU by writing



(a) Posterior of the torque and losses curve



(b) Mean posterior torque and loss curves



(c) Posterior response

Figure 2: Shown here are (a) Posterior obtained for the various points on the torque and loss curve (b) the Motor torque and losses curve obtained by taking the mean of each of the posterior distributions (c) 100 model response (blue lines) with 100 different draws of parameters from the posterior (Fig. 2a).

CUDA kernels that allow the simulation in parallel of multiple different vehicles with different input controls. A scaling analysis of the ROM on a NVIDIA A100 GPU can be seen in Fig 1b.

4 Future Work

We plan to improve our ROM's accuracy as well as answer several research questions. Specifically, (a) with the ROM fully calibrated, we plan to use it for state estimation studies to assess how the fidelity of the vehicle dynamics model affects the performance of a state estimator; (b) the GPU version of the model will be used to conduct large scale traffic simulations to perform Human In the Loop (HIL) and traffic congestion studies; (c) we plan to carry out analysis of variance over the parameter space – with the ability to run fast a large number of simulations on the GPU, we can evaluate *Sobol sensitivity indices* using Monte Carlo methods [8] to determine what parameters or combination of parameters affect the model response the most for a particular maneuver; and (d) study the inverse question, i.e. what maneuvers can help best identify model parameters of interest.

References

[1] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.

[2] A. Doucet, N. d. Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14.

[3] M. Manfred and W. Henning, *Dynamik der Kraftfahrzeuge*. Springer Vieweg Wiesbaden, 2015. [Online]. Available: https://doi.org/10.1007/978-3-658-05068-9

[4] G. Rill, "TMeasy – a handling tire model based on a three-dimensional slip approach," 01 2013.

[5] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-preserving Algorithms for Ordinary Differential Equations*. Springer Science & Business Media, 2006, vol. 31.

[6] A. Elmquist, A. Young, I. Mahajan, K. Fahey, A. Dashora, S. Ashokkumar, S. Caldararu, V. Freire, X. Xu, R. Serban, and D. Negrut, "A software toolkit and hardware platform for investigating and comparing robot autonomy algorithms in simulation and reality," *arXiv preprint arXiv:2206.06537*, 2022.

[7] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner, "Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC (with discussion)," *Bayesian Analysis*, vol. 16, no. 2, jun 2021. [Online]. Available: https://doi.org/10.1214%2F20-ba1221

[8] I. Dimov and R. Georgieva, "Monte carlo algorithms for evaluating sobol' sensitivity indices," *Mathematics and Computers in Simulation*, vol. 81, no. 3, pp. 506–514, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378475409002936