A Library of Lower Fidelity Dynamics Models (LFDMs) For On-Road Vehicle Dynamics Targeting Faster Than Real-Time Applications

Huzaifa Mustafa Unjhawala[1¶], **Ishaan Mahajan**[2], **Radu Serban**[3], and **Dan Negrut**[4]

1 PhD. Student, Mechanical Engineering, University of Wisconsin-Madison, Madison, USA **2** Undergraduate Student, Computer Science, University of Wisconsin-Madison, Madison, USA **3** Distinguished Scientist, University of Wisconsin-Madison, Madison, USA **4** Professor, University of Wisconsin-Madison, Madison, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 20 March 2024
Published: unpublished

Summary

Lower Fidelity Dynamic Models (LFDM) is a library of on-road wheeled vehicles that is written in C++ and CUDA and wrapped to Python using SWIG. Each of these on-road wheeled-vehicle dynamics models is described as a set of Ordinary Differential Equations (ODEs) that take a driver input - a normalized throttle between 0 and 1, a normalized steering between -1 and 1 (with -1 representing max steering toward a left turn), and a normalized braking input between 0 and 1, and subsequently advance the state of the vehicle (its position and velocity) forward in time.

In mathematical notation, these ODEs are of second order and are expressed as

$$\ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, \mathbf{P}) \ ,$$

where $\mathbf{x} \in \mathbb{R}^d$ is the d dimensional state of the vehicle, $\mathbf{u} \in \mathbb{R}^3$ contains the driver inputs, and $\mathbf{P} \in \mathbb{R}^k$ contains the k model parameters. The ODEs evolving the vehicle dynamics models are cast as an Initial Value Problem (IVP) by providing an initial state $\mathbf{x}$. The latter's solution is found using implicit or semi-implicit numerical integration methods.

The LFDM library contains three dynamic vehicle models, differentiated by their Degrees of Freedom (DoF) counts: 11 DoF, 18 DoF, and 24 DoF. Each can be run on the CPU or an NVIDIA GPU.

11 DoF Model: This is a foundational, single-track model with two wheels, primarily employed in controller design. In the literature, it also known as the bicycle model. It features a rigid chassis with three DoFs at the Center of Mass (CM) – yaw, lateral, and longitudinal. The model includes a kinematic driveline, integrating a map-based engine and torque converter, gearbox, and a differential that enables transfer of torque to the wheels. The vehicle model can use one of the two versions of the TMeasy non-linear tire model for traction-force generation. A steering input is converted to a front wheel angle through a steering map.

18 DoF Model: A step up above the simple 11 DoF model, this double-track model includes four wheels and introduces a roll DoF, enriching the chassis' dynamics. The model shares the driveline structure with the 11 DoF model but adds front and rear differentials for torque distribution across wheels (See Fig. Figure 1). The steering map in this model simultaneously changes the orientation of the front left and right wheels by the same angle. The other model subsystems are identical to those of the 11 DoF model.

24 DoF Model: This model incorporates all the features of the 18 DoF and extends the DoF count in order to predict vehicle heave and pitch motions. As such, it captures the chassis's six DoFs: lateral, longitudinal, vertical, yaw, roll, and pitch. Additionally, the model includes a DoF at each corner for vertical suspension travel. Subsystems, excluding the suspension aspect, are identical to the 18 DoF model.

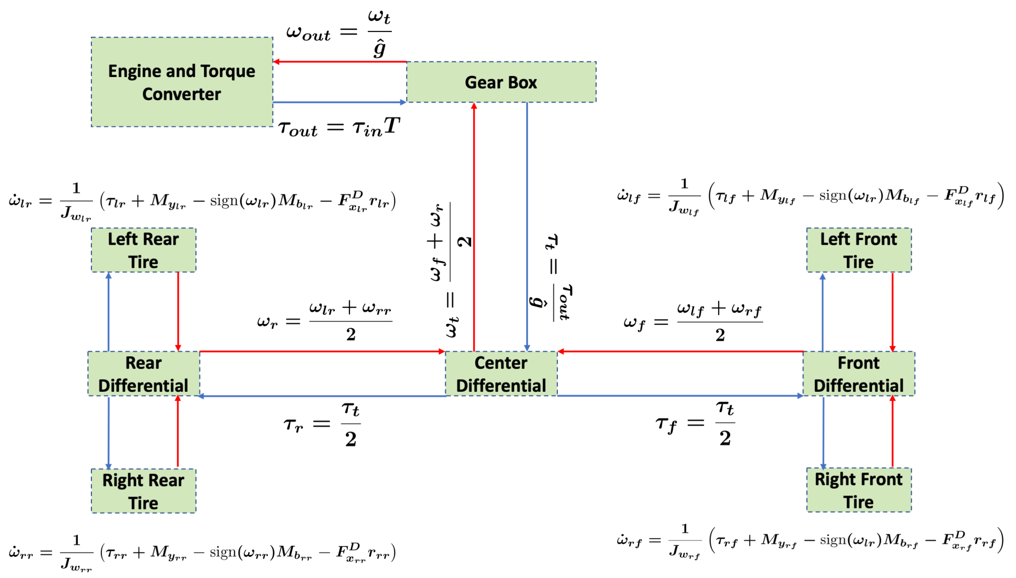

Figure 1: A flow chart of the torques (blue arrows) and angular velocities (red arrows) that are exchanged all across the drive line. An overview of the symbols is provided below.

Symbol	Description
J_{in}	Motor input shaft inertia ($kg \cdot m^2$)
T	Torque Ratio (-)
$\hat{g}$	Current gear ratio (-)
T	Torque Ratio (-)
η	Differential split (-)

These models have their own sets of customizable features, which are set based on user preferences. For example, the inclusion of the torque converter can be modified, and drivetrain configurations such as four-wheel drive (4WD), rear-wheel drive (RWD), or front-wheel drive (FWD) can be selected and adjusted through parameter configurations in a JSON file, offering flexibility to suit various simulation requirements.

In what follows, we discuss a Statement of need describing in what applications these models are most useful. In Section LFDM Accuracy, we present a comparison of the LFDM library's accuracy with the High-Fidelity Vehicle Model, Chrono::Vehicle (Serban et al., 2019). We then demonstrate in LFDM Speed and Scaling that the LFDMs, while closely matching the accuracy of Chrono::Vehicle, operate approximately 3000 times faster. Additionally, by utilizing the GPU version of the models, it is possible to simulate about 300,000 vehicles in real-time, i.e., simulating one second of dynamics for 300,000 vehicles takes only one real-world second. Further details on the model formulation are available in Chapter Two of (Unjhawala, 2023). Therein, the users can find the actual equations of motion, for each of the three models.

The Journal of Open Source Software

Statement of need

The open-source vehicle models in LFDM aim to provide accurate vehicle simulations that exceed real-time speeds ideal for state estimation, control, reinforcement learning, and traffic simulation, with or without a human in the loop. Informed by existing models in literature (Jin et al., 2019; Kong et al., 2015; Pepy et al., 2006), they go beyond the state of the art to introduce additional benefits tailored for faster-than-real-time applications.

1. In the domain literature, the focus is on simplistic single-track models with linear tires or fully kinematic models for their speed, despite a trade-off in accuracy. (Jiechao Liu & Ersal, 2016) found that while a double-track model with non-linear tires is more accurate, its Real Time Factor (RTF - the time require to simulate one second of vehicle dynamics) of 1 and higher, limits its use in Control stacks. The LFDM library, with efficient C++ coding, offers faster-than-real-time (RTF<1) double-track models with accurate non-linear tires and realistic subsystems, including drivelines, engines, and torque converters.

2. The existing state of the art for open-source vehicle models lacks documentation, hindering their use by researchers. Where open-source options exist, models of varying fidelity are scattered across multiple repositories. The LFDM library addresses this by providing a centralized source for researchers to access and select vehicle models of different fidelities, tailored to their specific speed-accuracy requirements and hardware capabilities.

3. To the best of our knowledge, there is currently no open-source software capable of executing large-scale, parallel simulations of on-road vehicle dynamics on GPUs. LFDM bridges this gap, facilitating the real-time simulation of nearly 300,000 vehicles. This capability significantly enhances the potential for large-scale reinforcement learning and comprehensive traffic simulations.

4. Vehicle models in literature typically use explicit numerical integration solvers for solving the associated equations of motion, but these display lackluster performance for non-linear, stiff ODEs (Ascher & Petzold, 1998). The LFDM library addresses this by offering two efficient time steppers: a semi-implicit solver with constant time-stepping for real-time simulations, and an implicit solver with adaptive time-stepping via Sundials (Hindmarsh et al., 2005). While the Sundials solver, due to its adaptive time-stepping nature, is slower and unsuitable for faster-than-real-time simulation, it ensures stability and supports Forward Sensitivity Analysis (FSA). Both solvers also generate system RHS Jacobians $\frac{\partial f}{\partial \mathbf{x}}$ and $\frac{\partial f}{\partial \mathbf{u}}$, which are the cornerstone of gradient-based Model Predictive Control (MPC) methods.

5. Commercial platforms like MSC Adams and MATLAB provide vehicle models of varying fidelities, complete with benchmarks. However, their closed-source nature limits the integration of these models into control stacks unless the same tools are employed throughout.

LFDM Accuracy

To assess the LFDM library's accuracy, we used the high-fidelity Chrono::Vehicle simulator to generate "ground-truth" data for its Sedan vehicle, which is a Chrono digital twin of the Audi A3. Each LFDM was then calibrated through a Bayesian Inference framework (Unjhawala et al., 2023) to approximate the dynamics (time evolution) of the Chrono::Vehicle digital twin. We conducted five throttle and steering maneuvers, varying throttle levels and steering directions, to calibrate the LFDMs. These maneuvers, ranging from 30 km/h to 65 km/h, helped fine-tune parameters for various turn speeds and calibrate engine and steering maps.

Post-calibration, we tested the LFDMs in two scenarios: a high-speed 90 km/h acceleration

The Journal of Open Source Software

and a 40 km/h double lane change, following ISO 3888-2 standards using a PID controller for the latter. The tests applied the same control inputs to both the Sedan and LFDMs. Results, illustrated in Fig.Figure 2 and Figure 3, reveal that the 24 DoF and 18 DoF models closely mimic the Sedan's dynamics, outperforming the commonly used 11 DoF model. However, in the straight-line acceleration test, the Chrono::Vehicle Sedan exhibited non-zero yaw and lateral velocities due to engine reaction torque, a detail not captured by the LFDMs. The LFDMs' accuracy was also compared against other vehicle types like the US Army's HMMWV, as detailed in Chapter 5 of (Unjhawala, 2023).

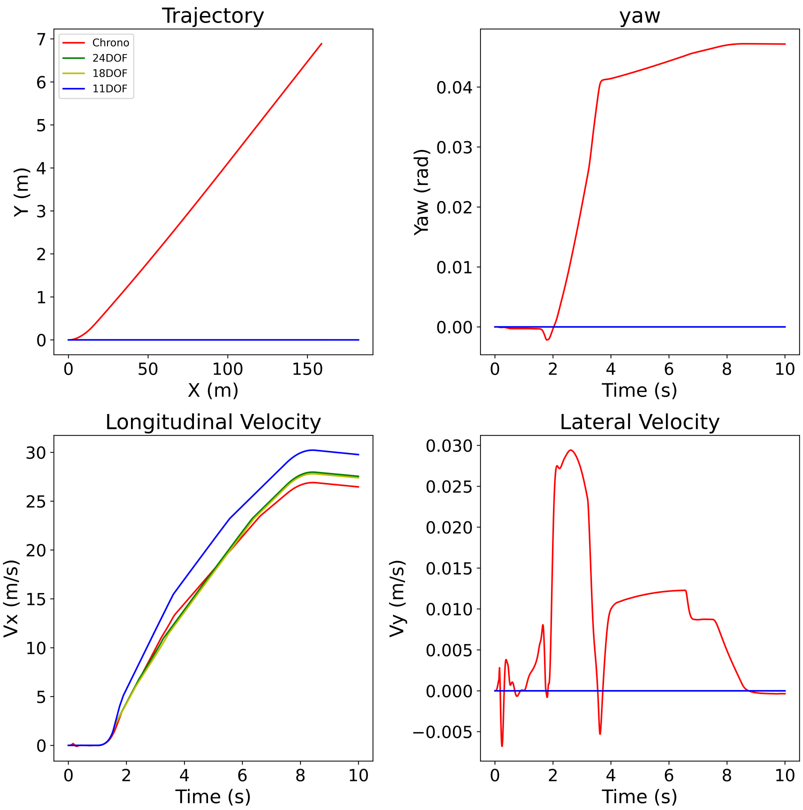

Figure 2: The LFDMs integrated using the semi-implicit solver with a time step of $1e^{-3}$ compared to a high-fidelity Chrono::Vehicle simulation of a Sedan integrated at a time step of $1e^{-4}$ for a high speed acceleration maneuver.

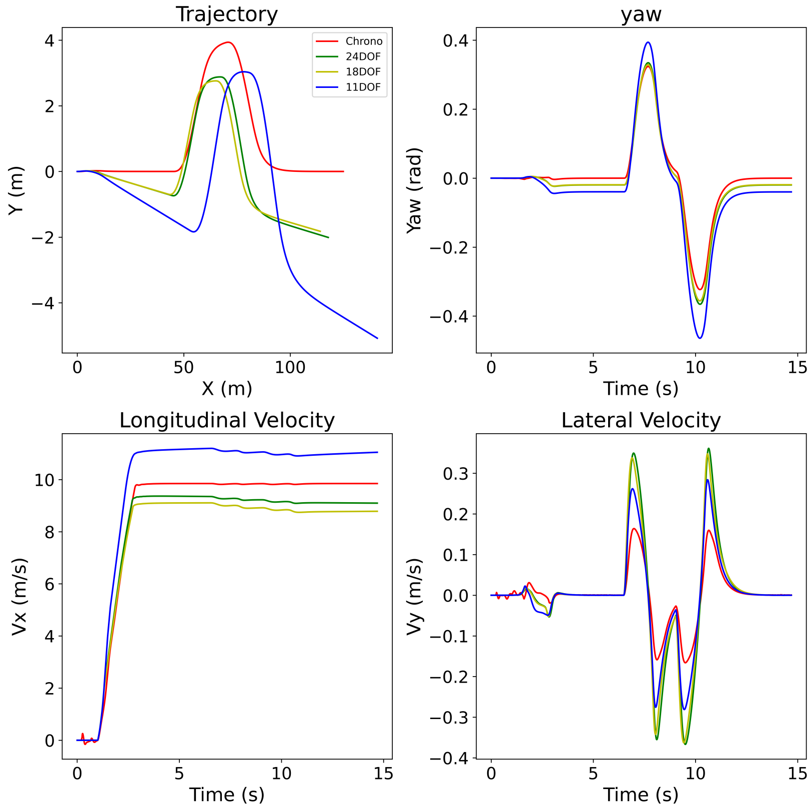

Figure 3: The LFDMs integrated using the semi-implicit solver with a time step of $1e^{-3}$ compared to a high-fidelity Chrono::Vehicle simulation of a Sedan integrated at a time step of $1e^{-4}$ for a ISO standard double lane change maneuver.

LFDM Speed and Scaling

To demonstrate the speed and accuracy of the LFDMs, we benchmarked them against Chrono::Vehicle. A 10-second acceleration maneuver was simulated for both, and the RTF was registered. Both systems used a $1e^{-3}$ time step, with the LFDMs employing a semi-implicit integrator and Chrono::Vehicle using its standard Differential Algebraic Equation (DAE) solver. The results, summarized in the table below, show that the LFDMs, when optimized with O3 and run on a 13th Gen Intel(R) Core(TM) i7-13700K, are at least 2000 times faster than real-time.

Model	Simulation time (ms)	Run time (ms)	RTF	$1/RTF$
Chrono::Vehicle	10,000	5134.29 ± 72	$5.1e^{-1}$	2
24 DoF	10,000	3.67 ± 0.0003	$3.6e^{-3}$	2720
18 DoF	10,000	2.6 ± 0.0006	$2.6e^{-3}$	3835
11 DoF	10,000	1.5 ± 0.0002	$1.5e^{-3}$	6461

Further, the GPU version of the LFDMs enables large-scale parallel simulation, which comes into play in Reinforcement Learning and traffic simulation. As shown in Fig. Figure 4, around 330,000 11DoF vehicle models can be simulated on an NVIDIA A100 GPU with an RTF of 1.

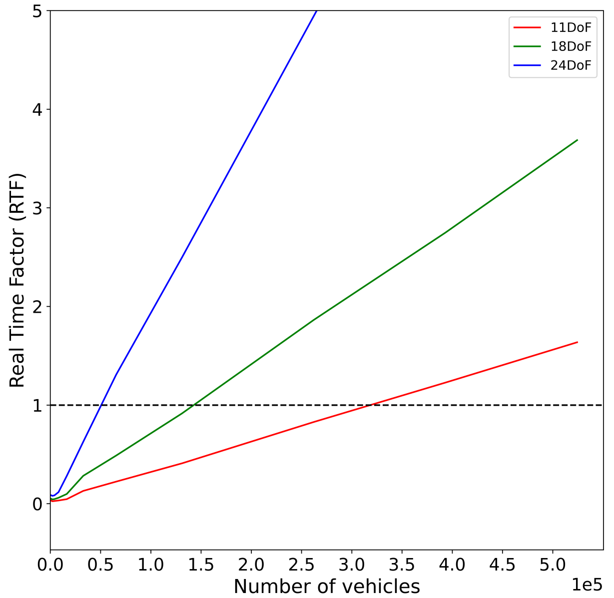

Figure 4: Scaling analysis of the GPU versions of the LFDMs shows that about 330,000 11 DoF vehicles can be simulated in Real-Time.

Acknowledgements

This work has been partially supported by NSF projects FW-HTF-R2222541 and CMMI2153855.

References

Ascher, U. M., & Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations*. https://api.semanticscholar.org/CorpusID:32366732

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, *31*(3), 363–396. https://doi.org/10.1145/1089014.1089020

Jiechao Liu, J. L. S., Paramsothy Jayakumar, & Ersal, T. (2016). A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles. *Vehicle System Dynamics*, *54*(11), 1629–1650. https://doi.org/10.1080/00423114.2016.1223863

Jin, X., Yin, G., & Chen, N. (2019). Advanced estimation techniques for vehicle system dynamic state: A survey. *Sensors*, *19*(19), 4289. https://doi.org/10.3390/s19194289

Kong, J., Pfeiffer, M., Schildbach, G., & Borrelli, F. (2015). Kinematic and dynamic vehicle models for autonomous driving control design. *2015 IEEE Intelligent Vehicles Symposium (IV)*, 1094–1099. https://doi.org/10.1109/IVS.2015.7225830

Pepy, R., Lambert, A., & Mounier, H. (2006). Path planning using a dynamic vehicle model. *2006 2nd International Conference on Information & Communication Technologies*, *1*, 781–786. https://doi.org/10.1109/ICTTA.2006.1684472

Serban, R., Taylor, M., Negrut, D., & Tasora, A. (2019). Chrono::Vehicle template-based ground vehicle modeling and simulation. *Intl. J. Veh. Performance*, *5*(1), 18–39.

Unjhawala, H. (2023). *A set of reduced order vehicle models for applications in machine learning, state estimation and control* [{M.S.} thesis, \url{https://uwmadison.box.com/s/zfy7zhg7c6i7nexfoo2r2r6fop5cu9it}]. University of Wisconsin–Madison.

Unjhawala, H., Zhang, R., Hu, W., Wu, J., Serban, R., & Negrut, D. (2023). Using a Bayesian-inference approach to calibrating models for simulation in robotics. *Journal of Computational and Nonlinear Dynamics*, *18*(6). https://doi.org/10.1115/1.4062199