32nd Electric Vehicle Symposium (EVS32)
Lyon, France, May 19-22, 2019

Model-Predictive Eco-Driving for Electrified Connected and Automated Vehicles

Dominik Karbowski, Jongryeol Jeong, Koen Elands, Iulian Dobrovolschi
Argonne National Laboratory, IL, United States, dkarbowski@anl.gov

Abstract

Automation and connectivity present a unique opportunity to improve transportation energy efficiency by optimizing the way vehicles drive, based on deep knowledge of the surrounding environment and communications between traffic agents. Vehicles can save energy through eco-driving, where the longitudinal speed is controlled in a way to minimize energy consumption. In this paper, we present a flexible control method to achieve eco-driving in electric vehicles (EV) and hybrid-electric vehicles (HEV). The method is based on model predictive control (MPC), in which the vehicle speed is optimized at each time step over a receding horizon, and at the following step, the horizon moves and optimization is run again, allowing a state feedback loop. The MPC algorithm presented here is easily adaptable to both EVs and HEVs, and is implemented in a novel Simulink-based simulation environment designed for eco-driving research. As a result, the algorithm interacts with models with complex dynamics, making it easier to implement in real-world systems. We present the performance of the algorithm in several real-world scenarios, and demonstrate approximately 7% energy savings in both vehicle cases.

Keywords: EV, HEV, connected, automated, optimization

1 Introduction

Sensors, as well as connectivity between a vehicle and other vehicles (V2V) or the infrastructure (V2I), provide information to the vehicle about its environment and future driving conditions. A vehicle with automated driving then uses that information to perform its mission and accomplish various objectives: improved safety, increased mobility, greater comfort, better use of travel time, increased road capacity (e.g., platooning), and others.

Automation and connectivity can also be used for eco-driving, which consists of adjusting vehicle speed to minimize energy consumption, for example, coasting to a red light or anticipating slopes. Eco-driving can be systemtically implemented in a connected and automated vehicle (CAV), thanks to the active velocity control and environmental awareness from sensors and V2X. There are various approaches to eco-driving for CAVs — Vahidi and Sciaretta provide an extensive state-of-the art review in [1] on how CAV technology can be used for energy saving. The eco-driving approach we propose in this paper controls the speed of an electrified vehicle in order to minimize the battery energy use of the controlled vehicle; no collaborative or centralized control is assumed.

A key requirement for the controller is for it to adapt to a changing environment and to the response of the vehicle itself, accounting for the imperfection of models assumed during optimization and the uncertainty of the environment. Model-predictive control (MPC) is a technique that accomplishes this requirement and is at the heart of the proposed eco-driving method. MPC uses the concept of receding horizon: at each time step, MPC computes the optimal command and state trajectories over an entire finite horizon (e.g., the next 20 seconds), but only applies the first step of the optimization. In the following time step, the horizon window moves one step further and the optimization is performed again. As a result, MPC performs the optimization at each time step with the most recent information about the state of the system, which creates a feedback loop that is critical to the stability of the system. The MPC concept is illustrated in figure 1. MPC is a well-established approach and is particularly well suited for linear and quadratic systems; in such cases, the optimization method applied at each time step is quadratic programming (QP), which is relatively fast and simple to implement. However, the linearity condition limits the number of control variables QP can optimize, and therefore leads to suboptimal results.

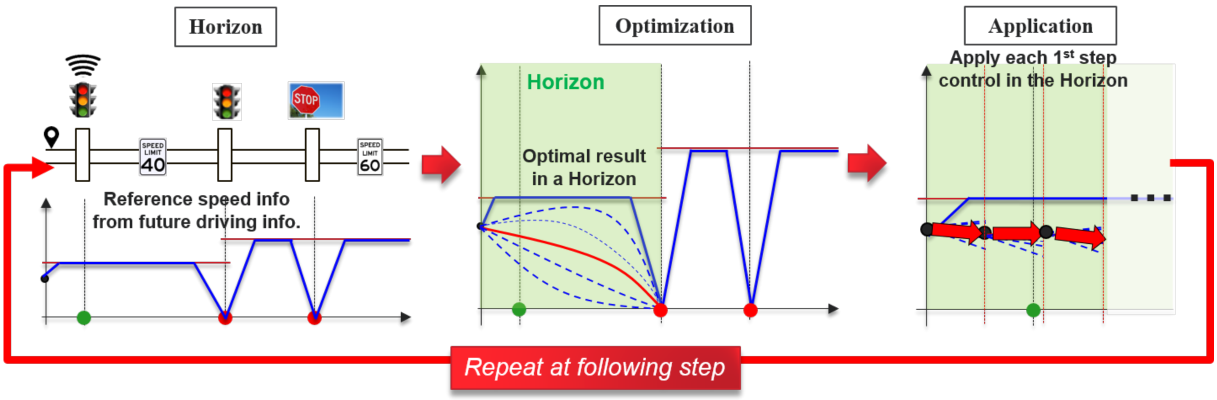

Figure 1: Concept of the MPC algorithm

Several examples of MPC applied to eco-driving exist in the literature [2, 3, 4, 5, 6]. We add several contributions to this body of work. First, we characterize the energy cost function as a function of wheel torque and vehicle speed, making the algorithm powertrain-independent, while demonstrating a simulation-based calibration method. Secondly, we strove to make the algorithm as implementable as possible, embedding it in a Simulink controller integrated with a high-fidelity dynamic model of the vehicle, and achieving a reasonable speed of execution. Thirdly, it is being integrated into RoadRunner [7, 8], a toolkit that allows us to run a broad range of realistic scenarios. Lastly, this control is designed to work for all "normal" driving situations: cruising, intersection approach and departure and car-following. We will introduce the latter functionality in future works.

In this paper, we first describe the MPC approach applied to eco-driving. We then introduce the simulation framework used to demonstrate the MPC controller, which is based on Autonomie [9] and on RoadRunner. Finally, we present the results of a case study.

2 Eco-driving with Model-Predictive Control (MPC)

2.1 Dynamic vehicle model

MPC uses a model to describe the dynamics of the system and to allow prediction of its future states. In this case, we use the standard dynamic vehicle model for longitudinal motion, including slope:

$$\sum F = m\frac{dv}{dt} = F_t - F_{rr} - F_D - F_g - F_b \tag{1}$$

where m is the mass of the vehicle, v is the linear speed, F_t, F_{rr}, F_D, F_g and F_b are respectively the traction force, rolling resistance force, air drag force, gravitational force and braking force. This leads to the following equation:

$$\frac{dv}{dt} = \frac{1}{m}(\frac{1}{R_w}T_w - \frac{1}{2}\rho A_f C_D v^2 - mg(C_{rr0} + C_{rr1}v + sin(\alpha)) - \frac{1}{R_w}T_b) \tag{2}$$

where T_w is the torque at the wheels, R_w is the radius of the wheel, g is the gravitational constant, C_{rr0} and C_{rr1} are static and dynamic rolling resistance coefficients respectively, ρ is the air density, A_f is the frontal surface of the car , C_D is the air drag coefficient of the car, α is the slope and T_b is the braking torque. Since (2) is not linear, we linearize it to:

$$\frac{dv}{dt} = Av(t) + Bu(t) + E\alpha(t) + f \tag{3}$$

where u is the command signal consisting of the torque at the wheels and the torque while braking ($u = [T_w, T_b]^\top$. A, B, E and f are constants that are function of v_0, the speed around which the linearization is performed. In order to make the simulation suitable for numerical evaluation on an on-board computer the continuous-time state space representation (3) is converted to a discrete-time representation. This is done with the following equations:

$$v(k+1) = A_d v(k) + B_d u(k) + E_d \alpha(k) + f_d \tag{4}$$

2.2 Energy consumption model

The main objective in our proposed optimization approach is to minimize the energy consumed by the vehicle. In the case of an EV, this is simply the battery energy. In the case of an HEV, there are two sources of power of different natures, electric P_e and chemical (fuel) P_f. We combine both powers into an equivalent $P_{eq} = \lambda P_e + P_f$ accounting for the significantly higher motor energy efficiency with an equivalence factor λ for the electric power. λ=2.5 based on the analysis of efficiency maps of the engine and electric motor.

The energy consumed E_{tot} is the sum over time of the battery power P_{tot}:

$$E_{tot}(T_w, v) = \int_{t_0}^{t_f} P_{tot}(T_w, v)dt \tag{5}$$

We assume that instantaneous power can be expressed in the form of a quadratic function of the wheel demand torque T_w, and v:

$$E_{tot}(T_w, v) = \int_{0}^{t_f} \left(k_1 T_w^2 + k_2 T_w + k_3 T_w v + k_4 v + k_0\right) ds \tag{6}$$

where k_0, ..., k_4 are constants depending the shape of the polynomial fit of the main instantaneous power (battery power for an EV and equivalent power for an HEV). We estimate these coefficients using Autonomie simulations of each vehicle over a large number of drive cycles. Figure 2 shows the contour plots of the polynomial fit functions for both EV and HEV. The accuracy of the second order polynomial fit is respectively 91% and 97%.

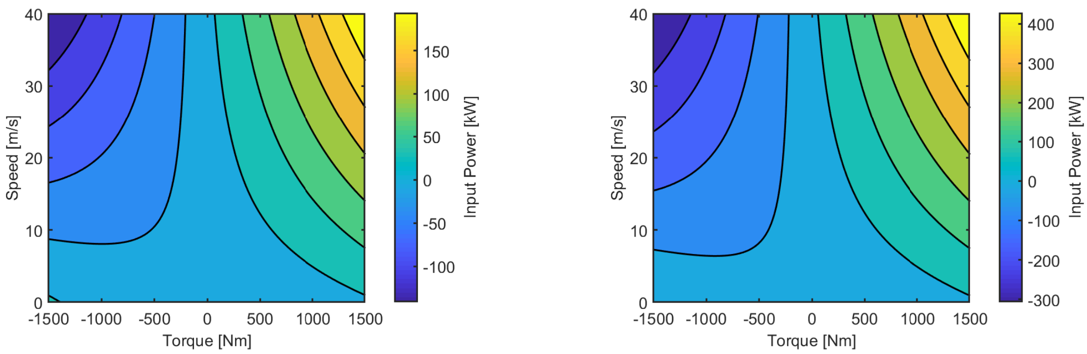

Figure 2: Power contour plot of the EV (left) and HEV (right)

2.3 Model reformulation with horizon

In the MPC framework, we assume the knowledge of a certain horizon discretized over N steps. The slope information and the speed limits over that timeframe are known, and a reference trajectory for that horizon is also known (detailed later). In MPC, we solve an optimal control problem (OCP) for that horizon. Let $v(k)$ be the state and $u(k)$ be the input at time instant k. Then the predicted state and input at time i is defined as $x_{i|k}$ and $u_{i|k}$ based on measured state at time k. The predicted state at time $i+1$ is derived using (3) and leads to the following equations:

$$v_{i+1|k} = A_d v_{i|k} + B_{d_{tot}} u_{i|k} + E_d \alpha_{i|k}, \forall i = 0, ..., N-1 \tag{7}$$

For a horizon length N, the predicted states are derived as:

$$\begin{aligned}
v_{1|k} &= A_d v_{0|k} + B_d u_{0|k} + E_d \alpha_{0|k} \\
v_{2|k} &= A_d v_{1|k} + B_d u_{1|k} + E_d \alpha_{1|k} \\
v_{3|k} &= A_d v_{2|k} + B_d u_{2|k} + E_d \alpha_{2|k} \\
&\vdots \\
v_{N|k} &= A_d v_{N-1|k} + B_d u_{N-1|k} + E_d \alpha_{N-1|k}
\end{aligned} \tag{8}$$

We can group the future states and predicted inputs over the prediction horizon into vectors X_k and U_k

$$X_k = [x_{1|k}, x_{2|k}, \dots, x_{N|k}]^\top \tag{9}$$

$$U_k = [u_{0|k}, u_{1|k}, \dots, u_{N-1|k}]^\top \tag{10}$$

By combining all the equations in (8), the future state X_k an be expressed as a function of the current state $v(k)$, and the predicted inputs U_k :

$$X_k = \Phi v(k) + \Gamma U_k + \Delta \Theta_k \tag{11}$$

where Φ, Γ and Δ are matrices derived from A_d, B_d, E_d. Φ and Γ are the prediction matrices for calculating the predicted sequence of states and inputs respectively and Δ is the matrix for calculating the influence of the grade (Θ_k) horizon on the predicted state sequence. Equation (11) forms the dynamic equation for the OCP that we are going to solve at each time step.

2.4 Cost function over the horizon

Two cost functions are present in this optimal control problem: the cost function that penalizes the error between state and input reference signal J_x and the cost function of the energy consumption over the horizon J_e. Adding these cost functions together will give:

$$J_{tot} = J_x + J_e \tag{12}$$

The cost function J_x is given with

$$J_x = (v(k) - r(k))^\top Q(v(k) - r(k)) + (X_k - R_k)^\top \Omega (X_k - R_k) + U_k^\top \Psi U_k \tag{13}$$

where $r(k)$ is the current reference speed signal, Q is the penalty on the difference between state and reference, Ω is the penalty matrix of the difference between predicted states and sequence of reference signals R_k and Ψ is the matrix that penalizes relatively big differences in input signals. After substituting (11) into (13) the cost function for the energy is written in a quadratic problem statement:

$$J_x = \frac{1}{2} U_k^\top G_x U_k + U_k^\top (F_x v(k) + H_x \Delta_k + V_x R_k) \tag{14}$$

We similarly reformulate the energy cost function, combining (6) and (11):

$$J_e = \frac{1}{2} U_k^\top G_e U_k + U_k^\top (F_e v(k) + H_e \Theta_k + V_e R_k + \Gamma^\top K_4 + K_2) \tag{15}$$

At each time step, the MPC algorithm will find the input/state U_k and X_k trajectories over the predicted horizon that that minimizes the cost function J_{tot} (12, 14, 15), while U_k and X_k are linked by dynamic equation (11).

2.5 Reference speed

As mentioned previously, one of the objectives of the MPC algorithm is to follow a state (speed) trajectory, which is generated through a rule-based logic. In cruising situations, the reference speed is the speed limit. For intersections with traffic lights, the logic is presented below.

2.5.1 Traffic light situations

The objective of the algorithm is to drive the vehicle at the highest possible velocity that will lead to crossing the intersection with a green traffic light. When the traffic light intersection is in range of the horizon the light can send information about its state and cycle to the vehicle (V2I communication). The algorithm calculates the state of the traffic light when the vehicle crosses the intersection while driving at the current speed. When it notices that the state of the light is red when crossing at that velocity, a reference signal will be made that will calculate the speed at which the car needs to drive in order to cross the intersection with green light. This is done every iteration for the whole horizon. First, the distance and time to intersection are calculated as shown below.

$$T_{isc} = \frac{D_{isc} - D_{veh}}{v} \tag{16}$$

where D_{isc} is the distance to intersection, D_{veh} is the current position of the vehicle and T_{isc} is the time to the intersection. The target velocity v_{target} can be determined using timing and distance information that the traffic light communicates to the car (V2I communication). This helps the vehicle avoid stopping at a red light then accelerating when it turns green, which causes a high amount of energy consumption as seen in [7]. The calculation of the target speed in several situations is computed as:

$$v_{target} = \begin{cases} v - \frac{2(D_{isc}-D_{veh})}{t_{ng}}, & \text{red and } T_{isc} \leq t_{ng} \\ v - \frac{2(D_{isc}-D_{veh})}{t_{nr}+t_r}, & \text{green and } T_{isc} \geq t_{nr} \\ v_{lim}, & \text{pass \& } D_{isc} < D_{nxt\ lim} \\ v_{nxt\ lim}, & \text{pass \& } D_{isc} > D_{nxt\ lim} \end{cases} \tag{17}$$

where t_{ng} is the time to the next green phase, t_{nr} is the time to next red phase, t_r is the red light duration, v_{lim} is the current speed limit, $v_{nxt\ lim}$ is the next speed limit and $D_{nxt\ lim}$ is the position of the next speed limit. For stopping scenarios, the control will be switched to the rule-based driving model in order to brake and reach a 0 velocity at the desired position. This rule-based manner of braking is implemented because the reference signal and the controller are distance-based instead of time-based.

2.6 Constraints

There are two types of constraints: on the state (speed) and on the input (demanded torques). Due to the limited availability of quadratic programming solvers embeddable in Simulink that would take non-constant state constraints (linear or quadratic), speed limits (upper and lower) are enforced through the reference speed. If we could use state constraints, we would relax the penalty Q which would result in a higher influence of the energy consumption the cost function J_e.

For input constraints, the maximum torque at the wheels is determined using a lookup table dependent on the current linear speed of the vehicle. The input constraints are given with:

$$\begin{aligned} &\underline{T_w} = \frac{-T_{m,max}(\omega_m)\gamma}{\eta} \\ &\underline{T_w} \leq T_w \leq \overline{T_w}(v) \\ &0 \leq T_b \leq \overline{T_b} \end{aligned} \tag{18}$$

where $T_{m,max}$ is the maximum torque the electric motor can deliver, dependent on the rotational velocity of the motor ω_m , η is the regenerative efficiency of the powertrain, and γ is the ratio between the torque at the electric motor and the torque at the wheels.

3 Simulation Framework

3.1 Powertrain model from Autonomie

We use Autonomie powertrain models to accurately simulate the energy consumption of the vehicle. Autonomie is a vehicle energy consumption and acceleration performance simulation tool developed at Argonne National Laboratory [9]. The powertrain models are written in Simulink language, and are forward-looking: a driver model computes acceleration and brake pedal commands based on a target speed (drive cycle) and current speed, which are then interpreted by controllers. Each plant subsystem follows an effort and flow structure, propagating efforts (e.g. torque, voltage) forward, and flows (e.g. speed, current) backwards.

Most subsystem models include test-based look-up tables that model the energy losses of the component, as well as dynamics and constraint blocks. It is important to note that the simulation model is of higher fidelity than the model used in the MPC controller.

3.2 Longitudinal dynamics simulation with RoadRunner

Predefined drive cycles are not well-suited to simulate CAVs, and our proposed control strategy in particular, as the vehicle itself dynamically decides its own speed based on its road environment and surrounding vehicles. As a result, we use RoadRunner [7, 8], a recently developed framework that can simulate multiple vehicles with full powertrain models and the interactions between vehicles and their environment. RoadRunner uses powertrain models from Autonomie, but adds new capabilities, such as multi-vehicle simulation, models of the road, causal models of human driving, V2X communications, and sensors. Figure 3 illustrates the steps in a typical RoadRunner use case. The user first defines a scenario: the route, the number of vehicles, the type of vehicles, and the type of CAV technology for each vehicle. RoadRunner then automatically builds the Simulink diagram, runs the simulation, and post-processes the results for the user to analyze.

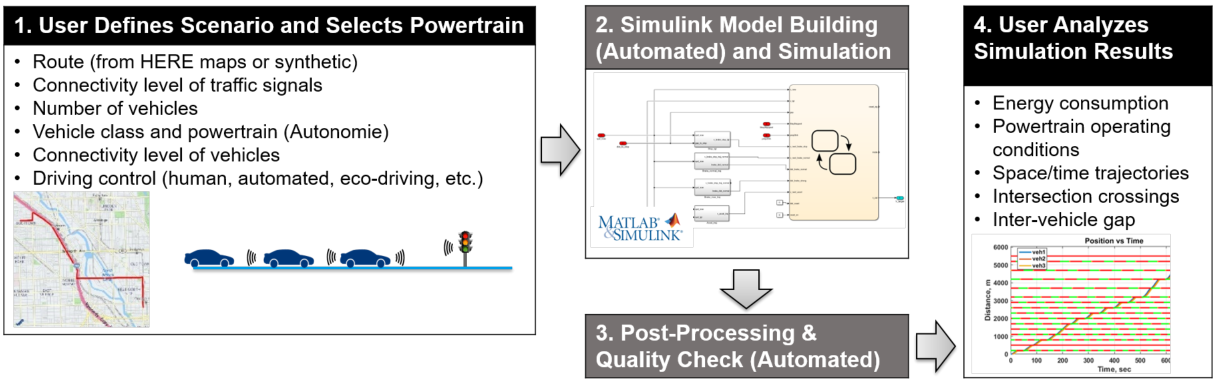

Figure 3: Overview of RoadRunner

Figure 4 shows a Simulink diagram generated from RoadRunner, featuring multiple vehicles. At the top level, an intersection block contains sub-blocks modeling each intersection along the simulated route: stop signs, traffic lights, connected or not. Each simulated vehicle has its own block composed of the vehicle itself as well as a signal router that propagates relevant information to the ego vehicle based on its position. Within each vehicle, a sensor/communication block models the real-world sensors and V2X communications by adding range, delays and other imperfections to the incoming "ideal" sensor data. The control block contains the longitudinal control, either a baseline human driver or the MPC controller presented in the previous sections. The controller interacts with the Autonomie powertrain in a close-loop.

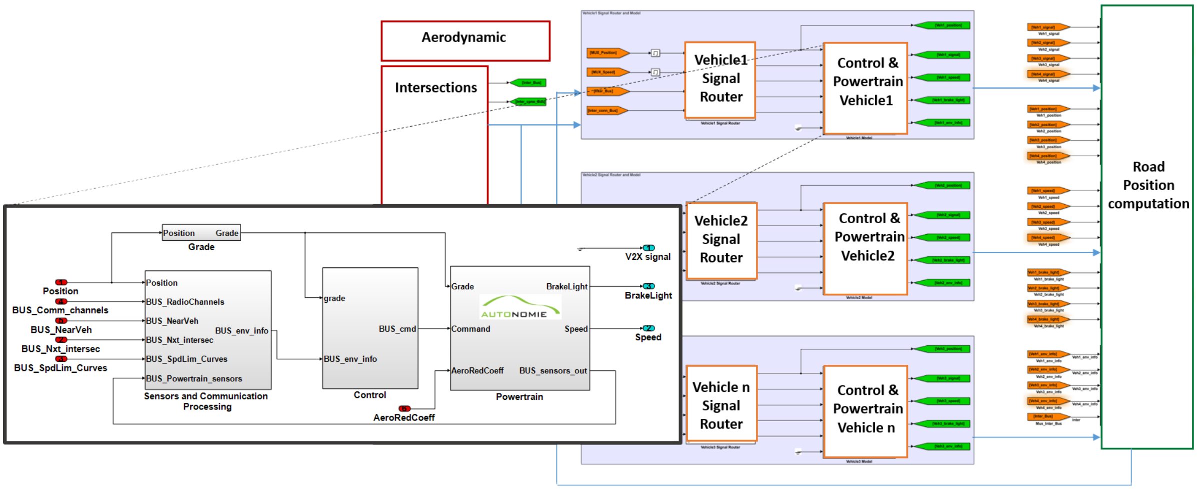

Figure 4: Top level and vehicle view of the Simulink diagram generated through RoadRunner

4 Results

4.1 Vehicle description

We applied the MPC algorithm to two vehicles modelled in Autonomie, both midsize cars. The EV was sized to deliver a 160 km driving range. The HEV is a power-split one mode HEV, very similar to the Toyota Prius. In the HEV case, we use the baseline energy management, as the MPC only controls the wheel torque demand. The engine on/off state, engine speed, power split and SOC management are decided through a rule-based controller validated on experimental data. The main specifications and architecture of both powertrains are shown in Table 1 and in Figure 5 below.

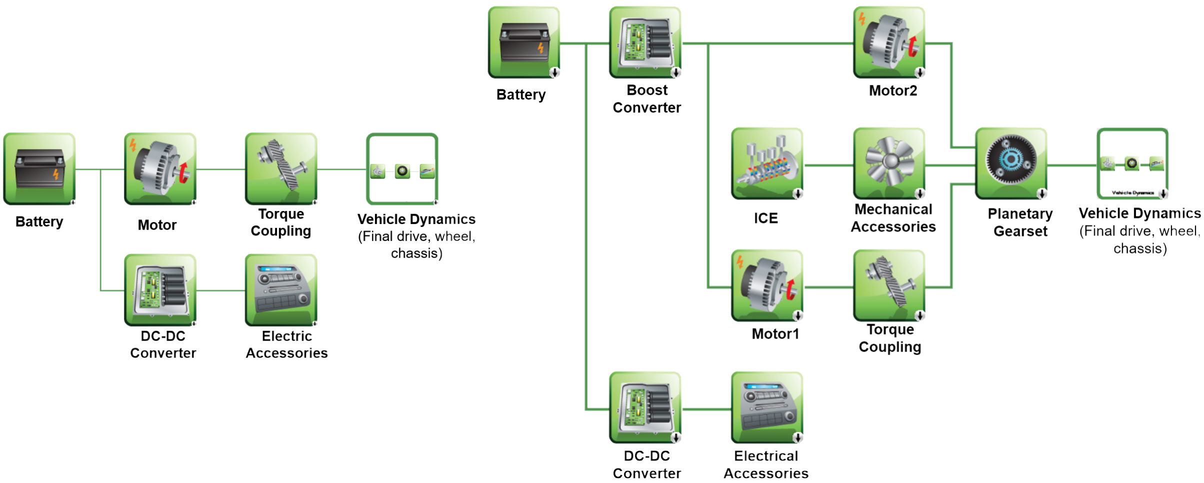

Figure 5: Electric vehicle (left) and HEV configuration (right) in Autonomie

4.2 Scenarios

We selected several scenarios to evaluate the performance of the eco-driving controller in RoadRunner. Figure 6 shows one of the selected routes for illustration purposes. The road attributes shown on the figure are extracted from HERE REST API. Table 2 summarizes the main features of each one of the routes. The routes include a significant amount of highway driving, in which the algorithm works as eco-cruise control. In each case, we simulate one vehicle featuring the MPC controller in an "open road" situation, i.e. without preceding vehicles that would constrain its speed. Inclusion of other such vehicles

Table 1: Main vehicle specifications

	EV	HEV
Mass	1784 kg	1669 kg
Aero/tire	$C_D A_f$=0.74 m^2, C_{rr}=0.08	$C_D A_f$=0.74 m^2, C_{rr}=0.08
Elec. Acc	460 W	240 W
Battery	Li-ion, 29 kWh, 156 kW	Ni-Mh, 1.4 kWh, 33 kW
Motor(s)	124 kW, η=90%	67 kW (M1), 52 kW (M2) η=90%
Engine		90 kW, η=38%

will be the subject of follow-up studies. In the baseline scenario, we use a driving model that drives at the speed limit and makes smooth transitions when speed limit changes and when arriving and leaving a stopping situation (due to a traffic light or a stop sign).

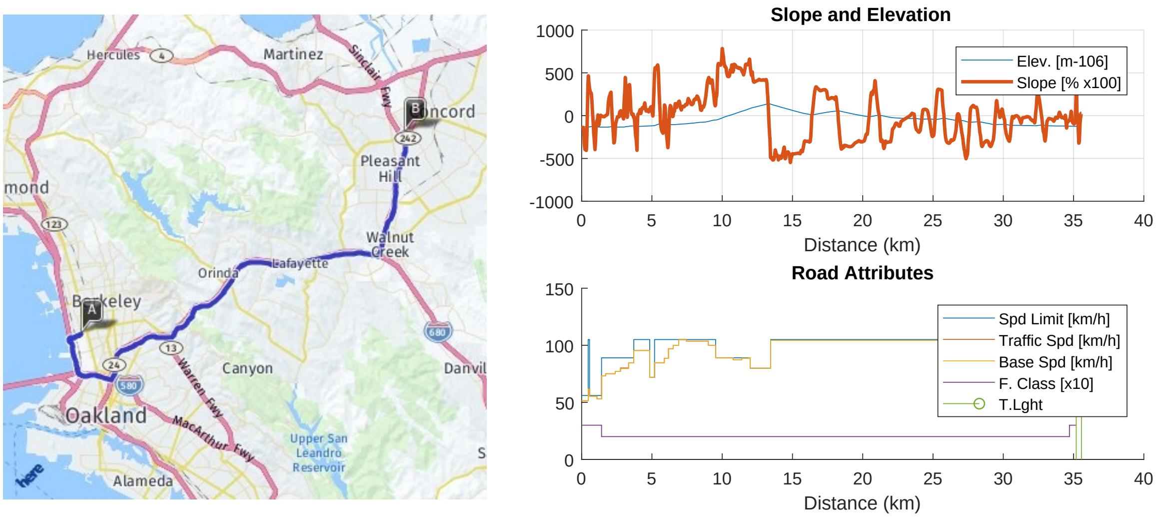

Figure 6: Overview of Route 1, Berkeley to Concord, California

Table 2: Route summary

	Distance (km)	Avg spd lim (km/h)	Traffic lights	Stop signs	Elev. gain (m)	Elev. loss (m)	Elev. net change (m)
Route 1	33.6	99.2	2	0	399	398	1
Route 2	46.8	82.2	17	0	257	216	41

4.3 Simulation results

Figure 7 shows a comparison of the speed profile between the baseline control and the MPC controller in the EV case for the route shown in Figure 6. The MPC controller anticipates changes in the speed limit and in the grade. As a result, we notice smoother accelerations and decelerations, as well as more fluctuations of the speed in the MPC case.

Tables 3 and 4 show the performance of the controller in terms of travel time and energy consumption for the EV and HEV. It is indeed important to compare these two often-competing metrics: energy savings cannot excessively penalize travel time, as this would likely lead to low use rates by drivers. This trade-off can be adjusted by tuning some of the MPC parameters, such as the terms penalizing deviation from the reference. For the routes simulated here, we notice a very marginal increase in travel time (2% or less) and significant energy savings (approximately 7%) for the EV. In the HEV case, the energy savings (6.5%) come at the cost of increased travel time (4 to 6 %).

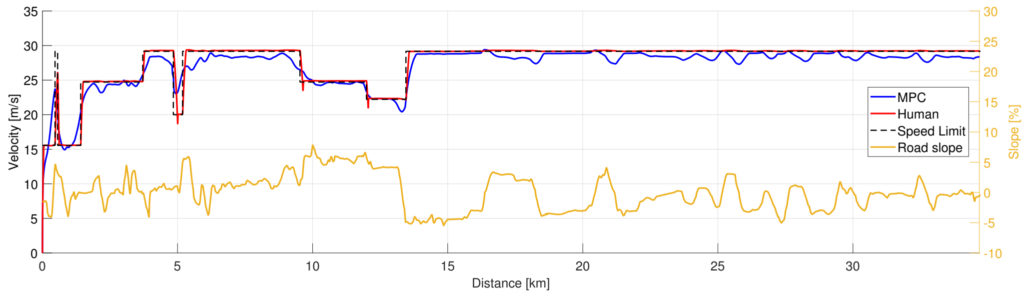

Figure 7: MPC vs baseline on the Route 1 scenario

Table 3: Travel times and energy consumption (EV)

	Travel time			Energy cons. (Wh/km)		
	base	MPC	% diff	base	MPC	% diff
Route 1	21m19s	21m45s	2.0%	139.1	128.5	-7.6%
Route 2	36m38s	36m39s	0.0%	120.9	112.8	-6.7%

Table 4: Travel times and energy consumption (HEV)

	Travel time			Fuel cons. (L/100km)			Δ SOC (%)	
	base	MPC	% diff	base	MPC	% diff	base	MPC
Route 1	21min26	22min50	6.5	4.13	3.87	-6.3	-11.8	-11.7
Route 2	36min41s	38min19s	4.4	4.15	3.878	-6.6	-3.1	-4.1

Tables 5 and 6 shed some light on the reasons behind such energy savings. In these tables, "brake energy" is the total braking energy at the wheels, including both regenerative braking and friction braking. The "regen recovery" is the percentage of the total braking energy that is recovered by the battery through regenerative braking. The most noticeable impact of the MPC controller is on the braking energy, which is decreased quite significantly. The MPC controller limits braking because braking, even regenerative, wastes kinetic energy either through friction or powertrain losses (especially motor losses). In addition, the MPC controller increases the regen recovery rate and reduces friction braking. We can also notice that it reduces the RMS current to and from the battery, which may have a beneficial impact on the battery's life.

Table 5: EV operating conditions

	Brake energy (Wh/km)		Motor efficiency (%)		Regen recovery (%)		RMS current (A)	
	base	MPC	base	MPC	base	MPC	base	MPC
Route1	30.4	18.2	86	85.2	76.2	80	204	187
Route2	20.5	11.1	83.7	83.10	75.6	74.4	212	195

Table 6: HEV operating conditions

	Prop. energy (Wh/km)		Brake energy (Wh/km)		ICE efficiency (%)		Regen recovery (%)	
	base	MPC	base	MPC	base	MPC	base	MPC
Route1	114.2	98.5	26.8	16.3	36.9	35.8	56.3	65
Route2	93	79.3	18	8	35.1	34.2	50.5	61

5 Conclusion

We have presented in this paper an eco-driving algorithm based on MPC and applied it to two CAVs, an EV and an HEV. We have demonstrated the effectiveness of the controller on simulated real-world routes, leading to 7% energy savings. The algorithm uses a universal energy model of the powertrain, a quadratic representation of the power source (fuel, battery, etc.) as a function of wheel torque and vehicle speed. Such model is a simplified one for most powertrains, which means that it does not necessarily lead to optimal results, but it also allows for easier implementation across a broad range of vehicles. This, coupled with the large number of powertrain models in Autonomie and the CAV scenario capabilities of RoadRunner, will enable to evaluate the energy benefits of eco-driving at a large scale. Future work will focus on adding more driving scenarios, including preceding vehicles, exploring calibration trade-offs and simulation with more powertrains and vehicle classes. We will also explore alternative solvers to better deal with state constraints.

Acknowledgments

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a DOE Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

This report and the work described were sponsored by the U.S. Department of Energy (DOE) Vehicle Technologies Office (VTO) under the Systems and Modeling for Accelerated Research in Transportation (SMART) Mobility Laboratory Consortium, an initiative of the Energy Efficient Mobility Systems (EEMS) Program. DOE Office of Energy Efficiency and Renewable Energy (EERE) manager David Anderson played important roles in establishing the project concept, advancing implementation, and providing ongoing guidance.

References

[1] Ardalan Vahidi and Antonio Sciarretta. Energy saving potentials of connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 95:822 – 843, 2018.

[2] Erik Hellstrom, Jan Åslund, and Lars Nielsen. Design of an efficient algorithm for fuel-optimal look-ahead control. *Control Engineering Practice (Special Issue on Automotive Control Applications, 2008 IFAC World Congress)*, 18(11):1318–1327, 2010.

[3] E. Kural and B. Aksun Güvenç. Model predictive adaptive cruise control. In *2010 IEEE International Conference on Systems Man and Cybernetics (SMC)*, page 1455–1461, Oct 2010.

[4] Marcus Kalabis and Prof Dr-Ing Steffen Müller. *A Model Predictive Approach for a Fuel Efficient Cruise Control System*, page 201–211. Gabler Verlag, 2012.

[5] Junbo Jing. *Vehicle Fuel Consumption Optimization using Model Predictive Control based on V2V communication*. PhD thesis, The Ohio State University, 2014.

[6] T. Schwickart, H. Voos, J. R. Hadji-Minaglou, and M. Darouach. A novel model-predictive cruise controller for electric vehicles and energy-efficient driving. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, page 1067–1072, Jul 2014.

[7] N. Kim, D. Karbowski, and A. Rousseau. A modeling framework for connectivity and automation co-simulation. April 2018.

[8] N. Kim, D. Karbowski, J. Jeong, and A. Rousseau. Simulation of heavy-duty vehicles in platooning scenarios. 2018.

[9] N. Kim, D. Karbowski, A. Rousseau, S. Halbach, and L. Michaels. Electric drive vehicle development and evaluation using system simulation. 47(3):7886–7891, August 2014.

Authors

Dominik Karbowski is Technical Manager of Intelligent Eco-Mobility at Argonne National Laboratory, leading Argonne's research on energy-efficient connected and automated vehicles. His research interests include automotive systems simulation, control theory, energy management, electrified powertrain design and optimization, as well as driver behavior modeling. Dominik is a major developer of simulation tools for energy-efficient vehicle research such as Autonomie and RoadRunner. Dominik holds a master of science in engineering from Mines ParisTech (France).

Jongryeol Jeong Jongryeol Jeong received his Ph.D. in Mechanical Engineering from Seoul National University in Seoul, Korea, in 2015. The main subject of his thesis was optimization of energy management and supervisory control of plug-in hybrid electric vehicle considering thermal aspects of vehicle components. He has been working in Argonne National Laboratory's Vehicle Modeling and Simulation Group since 2015. His research includes modeling various vehicle systems and components, validating simulation models, developing vehicle supervisory controls, and optimization of energy management.

Koen Elands is a graduate student in automotive technologies at the Eindhoven University of Technology, with a specialization in control systems. He is graduating a master's in 2019. In 2018, Koen worked as a Research Aide at Argonne National Laboratory to implement MPC for eco-driving.

Iulian Dobrovolschi received a master's degree in electrical engineering from the Eindhoven Technical University (Netherlands) in 2017. In 2016, Iulian worked as a Research Aide at Argonne National Laboratory. After graduation, Iulian joined TNO to work on energy market and energy production optimization.