

## Energy Consumption Prediction of a Vehicle along a User-Specified Real-World Trip

Dominik Karbowski,<sup>1,2</sup> Sylvain Pagerit,<sup>1</sup> Andrew Calkins<sup>3</sup>

<sup>1</sup>Argonne National Laboratory, <sup>2</sup>dkarbowski@anl.gov,

<sup>3</sup>NAVTEQ North America

---

### Abstract

Standard cycles provide an easy way to evaluate the energy consumption of vehicles, but it is the energy consumption that occurs on real-world trips that really matters to the driver and, to a larger extent, society. This study shows how digital maps and vehicle simulation tools can be used to estimate energy consumption on a real-world trip. The user (1) selects a trip in the mapping service ADAS-RP (Advanced Driver Assistance Systems Research Platform), (2) defines a vehicle model in the vehicle powertrain simulation tool Autonomie, and (3) runs and analyzes the simulation in that same tool. For each section of the trip, ADAS-RP provides various information that can include speed limits, historic data on traffic pattern speeds, the slopes of the routes, and the positions of stop signs and traffic lights. The first stage of processing this information is to schedule the stops and to create an intermediate speed target that takes those stops into account. The final driver demand speed includes transitions – accelerations and decelerations – between sections with different intermediate speed targets, or around stops. The ADAS-RP/Autonomie process is then used to compute the energy consumption of a hybrid electric vehicle and a conventional vehicle on 10 trips defined across the United States and Germany. The hybrid vehicle is more fuel efficient, especially on congested routes and routes with downhill slopes, the effect of which is analyzed in further detail.

Keywords: vehicle simulation, trip prediction, geographic information system (GIS)

---

### 1 Introduction

Most research on vehicle energy efficiency relies on predefined drive cycles as benchmark tests. Although this approach provides repeatable results and allows for easy comparisons, it ignores the relationships between the driver, the vehicle controller, and the environment.

This paper illustrates an approach that links the mapping and navigation service ADAS RP [1] and the advanced vehicle simulation tool Autonomie [2]. The goal is to predict the amount of energy consumed on a user-specified route. An added benefit is that the approach creates an innovative framework for various research topics that rely on real-world trips: “green” routing, hybrid control optimization, fleet management, etc.

Applications of the approach could be especially significant in the case of hybrid electric vehicles (HEVs). The added degree of freedom from a second source of power allows for greater flexibility in the design of the controller. For example, several studies show that the reduction in fuel consumption by HEVs can be higher on roads with slopes if future road information is made available to the controller.

The main challenge in this study is to create a speed and grade profile for a trip when those data are limited. The challenge is very similar to the online future-trip prediction problem.

Future route prediction is a promising research topic because such data are essential input for optimal controllers for advanced powertrains like HEVs. Dynamic programming ([3], [4]) and the Pontryagin minimization principle ([5], [6]) are the two main control theory techniques used for

advanced powertrains. Both require full knowledge of the trip profile ahead to compute the optimal control law. Some heuristically optimized controls also rely on trip prediction [7].

There have been several real-world implementations of road prediction with energy efficiency in mind ([8]–[11]). One example is an experiment by Nissan on a hybrid Tino [8]. Its controller uses a combination of road classification, grade, congestion level, and historic data to reduce the overall fuel consumption of the vehicle. Road prediction has also been explored for heavy-duty applications, mainly line-haul trucks ([10], [11]). Adaptive cruise control uses the knowledge of the slopes ahead to adapt the target speed and gear, thus leading to non-negligible fuel savings. The full long-term horizon, however, is not fully computed; for example, speed is predicted for a very short time-frame, if at all.

Predicting future driving is possible and can be done in various ways. One is to predict a future trip based on past trips of the same driver ([12]–[14]). After several trips are logged and their patterns analyzed, it is possible to find out which trip the user is going to make. This approach was developed by Krumm (Microsoft) and Froelich (University of Washington) ([13]–[14]).

Another technique relies on digital maps and knowledge of the destination. A very simple approach can rely on known speed limits and topographic maps [15] to, for example, compute an electric car range. In a more complex approach, a navigation engine can compute the most probable path to the destination; then with the content-rich digital map, the speed on each subsection of the road can be estimated. In [16], Minett et al. used ADAS-RP, developed by NAVTEQ, to compute an estimated speed trace and fuel consumption by using a backward-looking vehicle model. We use a similar approach in our study, but we added to the trip prediction details that have a major impact on energy consumption, such as stop signs, traffic lights, and grade. We also made it a process within Autonomie, a licensable, forward-looking simulation tool. The user can simulate any type of vehicle defined in Autonomie on any type of trip defined in ADAS-RP.

This paper first presents the tools that we linked together: ADAS-RP and Autonomie. It then explains the two stages behind the final speed target computation. Finally, it illustrates how this

process can be used to benchmark a hybrid vehicle against a conventional one.

## 2 User Interfaces

### 2.1 Autonomie

The vehicles are simulated in Autonomie, which was developed by Argonne National Laboratory (Figure 1) [2]. It is the main vehicle simulation tool used in the U.S. Department of Energy’s FreedomCAR and Vehicle Technologies Program. It has been used in numerous studies to give the U.S. government guidance for future research. More than 140 companies and research entities, including major automotive companies and suppliers, also use Autonomie to support advanced vehicle development programs.

Autonomie is a software package with a user-friendly interface that allows fast selection of powertrain configurations and component models, initialization data and processing files. It includes a wide range of pre-defined vehicles, both light and heavy-duty. All models and calculations are based on Matlab and Simulink: a vehicle is built in Simulink based on the selections made in the interface, and it is similarly initialized.

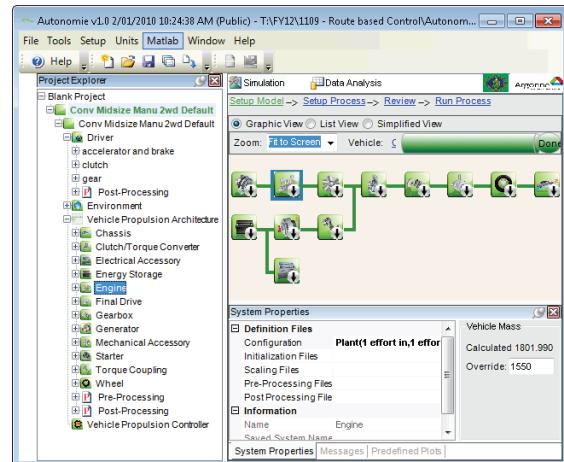


Figure 1: Autonomie GUI

The resulting vehicle model is a forward-looking model: a driver model sends a pedal signal to the supervisory control of the vehicle, which in turn sends a torque demand to the engine and other power sources. The components use an effort/flow topology, inspired by the Bond graph theory: an effort (e.g. torque) is propagated forward until it is integrated to compute a flow (e.g. speed) which is itself propagated backwards.

The plant models (engine, transmission, etc.) are generally map-based, combining accuracy and short execution time. Vehicles and individual components were validated based on test data.

## 2.2 ADAS RP

ADAS RP (Advanced Driver Assistance Systems Research Platform) is a software framework developed by NAVTEQ [1]. It is used to develop prototypes of applications that use positioning and maps for a wide range of applications, from eco-routing to headlamp orientation in turns.

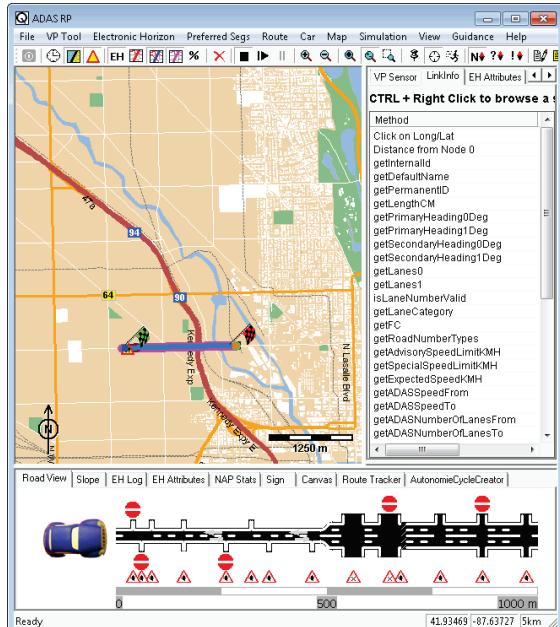


Figure 2: ADAS RP

ADAS RP give the developer the access to the road geographical data collected by NAVTEQ. Thanks to the multiple plug-ins, these data can be used for the specific needs of the application.

NAVTEQ maps provide a highly accurate representation of the detailed road network. Numerous attributes are recorded by geographic analysts who continuously drive on the road network, ensuring a verified and up-to-date database. The road network is segmented in small sections, or links, which can have up to

260 attributes, including vehicle speed limits, slopes, and number of lanes. NAVTEQ maps also include the positions and types of road signs, particularly stop signs and traffic lights.

NAVTEQ is also a major provider of traffic information, thanks to its extensive network of proprietary probes and data processing capabilities. ADAS RP leverages this expertise by including traffic patterns on major roads.

In a typical utilization, the user can select a starting point and destination on a map integrated in the graphical user interface (GUI). The internal map engine then computes the most likely route, and it is possible to see the attributes of each link.

## 2.3 Trip definition to vehicle simulation

From a user point of view, simulating a vehicle in Autonomie on a trip defined in ADAS RP is done in four steps (Figure 3). The user first defines the trip by selecting the origin and destination. This is done by using a visual map that is easy to navigate and zoom in and out on.

An Autonomie plug-in for ADAS RP was created and is displayed as a tab in the ADAS RP GUI. From that tab, the user can export the cycle, which saves relevant data in a Comma-Separated file (CSV).

The user then switches to the Autonomie GUI. After defining the vehicle to simulate, he chooses an “ADAS RP” process, and selects the CSV file corresponding to the trip he wants to simulate.

As with any other cycle, the user can launch the simulation and then analyze the results.

When the CSV file containing the road data is being selected in Autonomie, several algorithms process the raw ADAS data into a usable form. First, the processing of the road information per se occurs: the Road Data Processing (RDP). This is followed by running the off-line part of the distance-based driver model, in which the vehicle speed target is computed. Both processes are described in Sections 3 and 4, respectively.

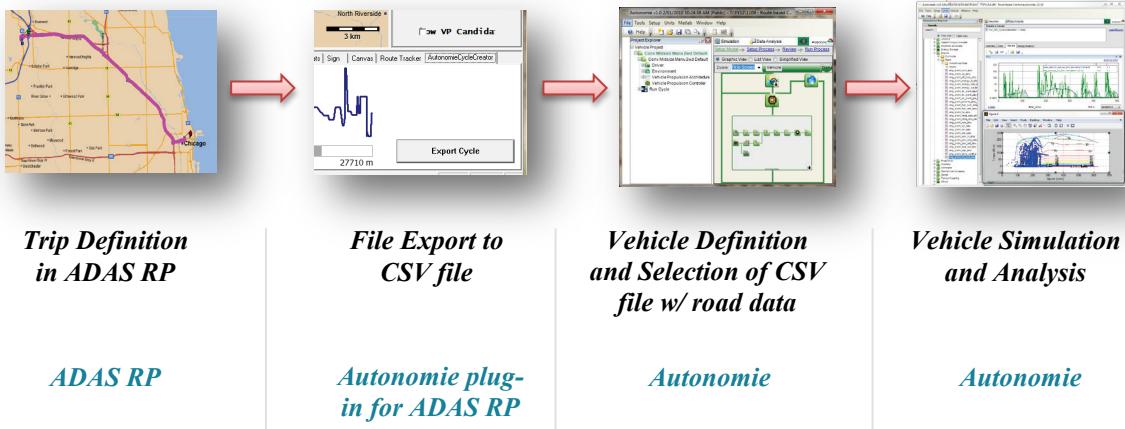


Figure 3: Work flow for user simulating a vehicle in Autonomie over a trip defined in ADAS RP

### 3 Road Data Processing (RDP)

#### 3.1 Relevant road attributes

The data produced by ADAS RP contain a lot of useful information about the road travelled. The most important is speed, which comes from various signals, depending on the level of details contained in the map. The list below ranks them by increasing level of fidelity (i.e., how well they compare to the actual real-world speed on the link):

- Expected speed: typical speed on links of the same category;
- Speed limit: self-explanatory;
- Special speed limit: applies only at particular times (e.g., school days); and
- Traffic pattern speed: average traffic speed that is based on historical data and depends on the time and day of the week.

The approximate speed is computed from aggregating the speed signals and keeping the value from the most accurate available attribute. NAVTEQ maps also contain a spline representation of the road, so the high-fidelity slope value is generally available at a higher space resolution than are most attributes.

To accurately model real-world driving, it is necessary to include stops. There are several reasons a vehicle stops, but we identified two major ones: stop signs and traffic lights. The raw data from ADAS RP include the positions of these two types of road signs.

#### 3.2 Rationale for processing raw data

There are several limitations to the raw data that makes it not directly usable for a realistic vehicle simulation.

First of all, attributes of interest are not available on all links. Secondly, there is no information on the stopping time, but simply location of road signs. Finally the traffic pattern speed, when available, is an average traffic speed, and as such already takes into account any stops along the way. The goal of the processing is to create a target speed for each of the links as well as the location and duration of stops. This information will then be processed to generate the driver demand speed, which is described in Section 4.

To create this information, some basic assumptions about how the vehicle is actually driven are made. In a first approximation, it is assumed that the trip is composed of a continuous succession of constant accelerations, constant speeds, and constant decelerations (Figure 4).

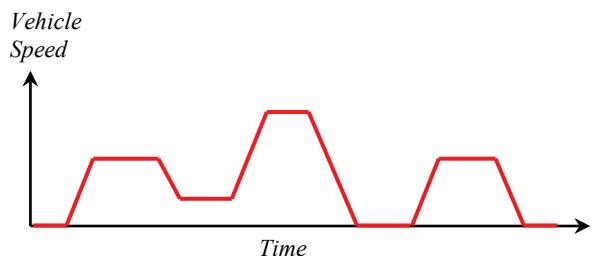


Figure 4: Example of assumed speed profile

The output of the RDP, the target speed, will actually be simpler than the profile shown in Figure 4. The acceleration and deceleration parts, while taken into account for average speed adjustments, will not be part of the output. The target speed will be composed of constant speed sections (“stairs”), while the transitions will be added when generating the driver demand speed. The RDP has several steps:

- Data normalization/formatting,
- Stop scheduling,
- Division into subsections, and
- Target speed computation.

### 3.3 Stop schedule

Two types of traffic signals lead to stopping: stop signs and traffic lights. The total stopping time depends on the signal (green/red time for lights, 4-way or 2-way stops, etc.) and also on the level of congestion, or level of service (LOS). In this study, we use a simple model, in which the total stopping time depends on LOS (Table 1). To do so we identify three levels of service – free flow, congestion, and heavy-congestion – based on the difference between the speed limit and the traffic pattern speed. For stops signs, there is a longer stopping time with higher congestion.

Table 1: Stopping time for various levels of service

LOS	Stop sign time	Red light time	Green light time
Free-flow	1	30 s	30 s
Congested	2	35 s	25 s
Heavy-congested	3	40 s	20 s

For traffic lights, we assume all traffic lights have the same period (60 s). However, the stopping time is longer when congestion levels are higher. We use the approximate speed to estimate the time at which the vehicle will arrive at the traffic light. Meanwhile, each traffic light is randomly initialized, and the alternation of red and green is computed. The vehicle will either wait for the red light to turn green or not stop at all, depending on the state of the light at the estimated time of arrival.

We can therefore create a stop schedule (i.e., establish where the vehicle needs to stop, and for how long).

### 3.4 Division in sections and subsections

Not all road links have the same number of attributes. In particular, the traffic pattern speed is available only on the major roads. The processing of a section with or without a traffic pattern speed is different, so the first task is to separate the sections depending on whether they have that attribute or not. This is shown in Figure 5. Each section with traffic (which can be made up of several links) is itself divided into constant speed sections; each of these sections and subsections has a defined set of attributes; they are defined by using structure arrays in Matlab.

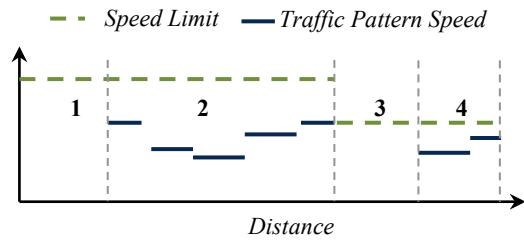


Figure 5: Division of the trip in sections with and without traffic pattern speed

### 3.5 Computation of intermediate speed target

Traffic pattern speed is an average speed, in which any stops are already incorporated. To reconstruct the real-world pattern, we assume a simple speed profile. In the simple case of a section with a constant speed, a constant deceleration, and a stop period, it is possible to compute the target speed  $v_T$ .

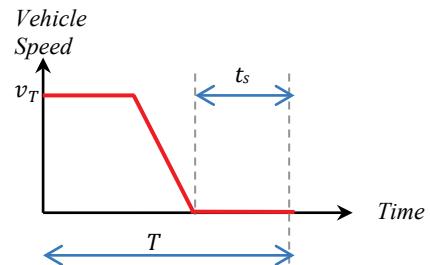


Figure 6: Example of a simple speed profile

In the example road section of Figure 6,  $T$  is the section's duration,  $t_s$  is the stop's duration, and  $L$  is the section's length. By using kinematic equations, the intermediate target speed  $v_T$  can be found through Equation 1:

$$\frac{v_T^2}{2d} - (T - t_s)v_T + L = 0 \quad (1)$$

Equation 2 is a generalized case where there are  $n_a$  accelerations,  $n_d$  decelerations, and  $N_s$  stops, each of duration  $t_{s,k}$ , within the same section:

$$v_T^2 \left( \frac{n_a}{2a} + \frac{n_d}{2d} \right) - \left( T - \sum_{k=1}^{N_s} t_{s,k} \right) v_T + L = 0 \quad (2)$$

Equation 2 does not always have a solution. When this is the case, the section with no solution is combined with an adjacent one; the lengths and times are respectively added together; and equation (2) is applied to that bigger section. That process is continued until a section is big enough to have solutions to the problem.

Figure 7 shows RDP inputs (traffic pattern speed, stop positions) and outputs (target speed, stop durations) for a trip defined in ADAS RP.

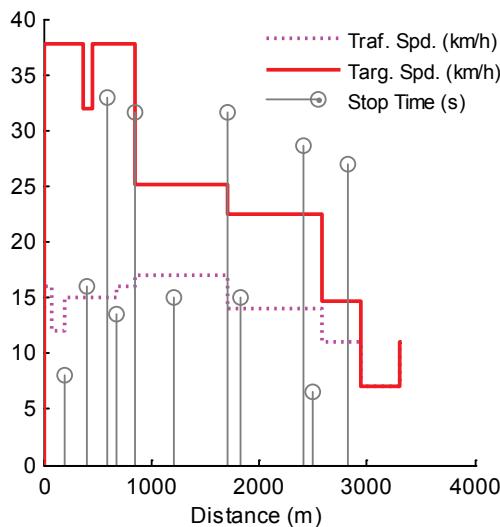


Figure 7: Traffic speed (from ADAS RP), target speed and stop durations for a sample trip

## 4 Computation of driver demand speed

The next step in the process is to transform the target speed that was computed previously into a usable form for the Autonomie driver, the demand speed. The Autonomie driver model computes a torque demand from three terms:

- Baseline torque demand corresponding to the rolling and aerodynamic losses,
- Acceleration/deceleration torque using the derivative of the demand speed, and

- Corrective torque from the comparison of demand speed and current speed in a proportional-integral control.

If a major discontinuity occurs, the corrective torque will jump suddenly, akin to sudden full throttle acceleration or full braking. In most real-world situations, the human driver anticipates such increases and decreases in speed target and uses his experience to apply accelerator or brake commands moderately. The “stairs” coming from the previous step therefore need to smoothed out and incorporate the transition between the various speed levels.

### 4.1 Division in sections

The trip is again divided into sections, so that stops and major discontinuities in speed are located at their extremities (Figure 8). In other words, there is no stop or sudden speed difference in the middle of a section.

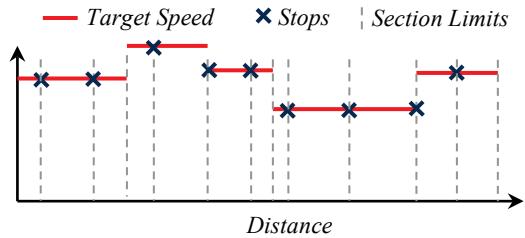


Figure 8: Example of division of the target speed

### 4.2 Computation of speed profiles for each section

Figure 9 illustrates intermediate target speed  $v_T$  and final demand speed  $v_D$  after inclusion of transitions. The schematic trip is divided into four sections,  $S_1$ – $S_4$ . If there is a sudden increase in target speed (here at  $d_1$ ), demand and target speeds will be the same in the section ( $S_1$ ) before the increase:  $v_D(d_1) = v_T(S_1)$ . In the section ( $S_2$ ) after the target speed increase,  $v_D$  will continuously increase from  $v_T(S_1)$  to  $v_T(S_2)$ , until the target speed of  $S_2$ ,  $v_T(S_2)$ , is reached.

If there is a sudden target speed decrease (here at  $d_2$ ), the demand speed after the decrease, in section  $S_3$ , will already be at the new lower target speed:  $v_D(d_2) = v_T(S_3)$ . In  $S_2$ , before  $d_2$ , the demand speed will start to continuously decrease from  $v_T(S_2)$  so that it reaches  $v_T(S_3)$  at the end of  $S_2$ .

Another example of transition is around  $d_3$ : a deceleration is added just before the stop at  $d_3$  and an acceleration occurs right after it.

After such processing, the demand speed, the grade, and the stop schedule can be fed into the Autonomie distance-based driver. In that driver model, the speed demand is updated on the basis of the actual distance achieved and not the time. Also there is a stop schedule that gives the positions and durations of stops as a function of distance. Whenever the vehicle reaches the position of a stop, the speed demand zeroes out, and the vehicle has to wait for the duration of the stop before a positive target speed is requested again.

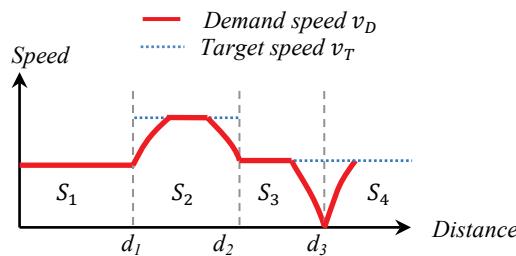


Figure 9: Computation of demand speed from target speed (schematic)

## 5 Simulation Results

### 5.1 Definition of sample trips

Table 2 – Presentation of trips

Trip	Distance (km)	Description	Composition*			
			U	UH	EH	ER
Boston	28.1	MIT to Salem	50	35	0	15
LA	14.8	Staples Center to Beverly Hills	100	0	0	0
SF	26.5	Fort Mason to San Rafael	10	90	0	0
Chicago1	40.1	Argonne to Navteq	20	80	0	0
Chicago2	3.3	Wicker Park to Gold Coast	100	0	0	0
Chicago3	6.7	Bucktown to Loop	60	40	0	0
WashDC1	11.7	DOE to Alexandria	20	80	0	0
WashDC2	3.8	DOE to GWU	100	0	0	0
Detroit	21.3	GM HQ to Ford museum	35	65	0	0
Munich	35.7	BMW HQ to Possenhofen	10	40	30	20

\* Decomposition of the trip in four types of roads: U (urban), UH (Urban Highway), EH (Extra-Urban Highway), and ER (Other Extra-Urban Roads). Qualitative information only.

To evaluate the process, 10 sample cycles were defined in ADAS-RP and used in the following studies. They are within or around major urban centers in the United States and Germany: Boston, San Francisco, Los Angeles, Chicago, Washington D.C., Detroit, and Munich. Those cycles are described in Table 2.

### 5.2 Trip model

An example of simulated trip is shown in Figure 10. The “Traffic” signal is the traffic pattern speed directly extracted from ADAS RP. The “Target” signal is the result of the RDP: it does not include any stops, but it does into account for the extra time needed for accelerations and decelerations and stops; hence, its value is higher than that of the target speed. The “Simulated” signal is the actual vehicle speed from Autonomie simulation. Simulated and target speeds differ by the numerous stops in the simulated speed.

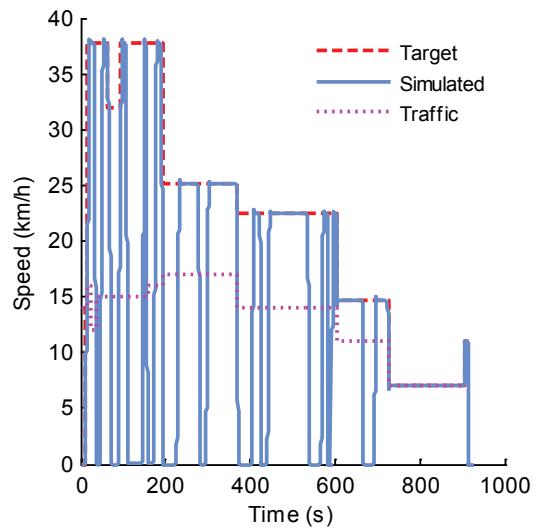


Figure 10: Simulated trip Chicago2

Figure 11 is a close-up of that same trip, and it highlights the way transitions occur. One example of transition, at  $t = 65$  s, is a speed reduction. The target speed decreases ahead of the change, so that it is never higher than the target speed. The simulated speed follows it closely. At  $t = 73$  s, there is a stop that lasts for 15 s. It is anticipated with a deceleration from the target speed to zero. After the stop time has elapsed, the target speed rises back to a strictly positive value (4 km/h). The discontinuity is necessary to make the vehicle start moving. During the acceleration, the vehicle follows the target speed, and the discrepancies are due to the natural lag of the controller (reaction time) as well as powertrain dynamics, such as gear shifts.

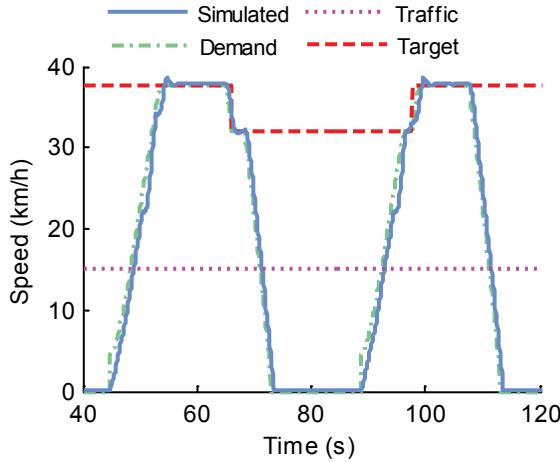


Figure 11: Speed changes, decelerations, accelerations, and stops in simulation

Figure 12 compares the travel time for each trip, computed directly from raw ADAS-RP signals and actual travel time from simulation. The two durations are generally close but not the same.

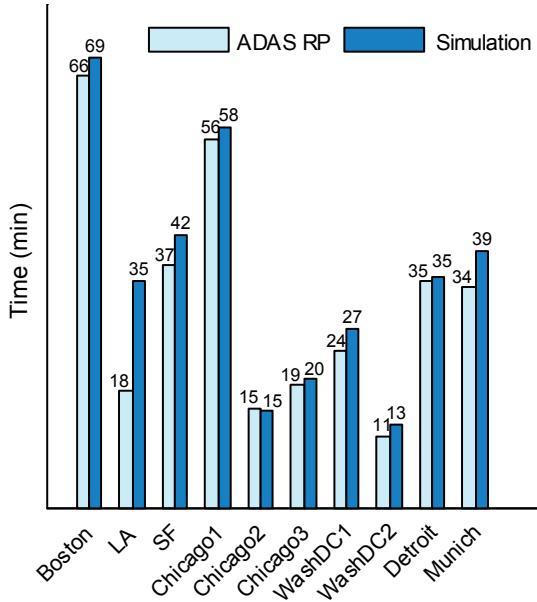


Figure 12: Comparison of trip time from ADAS RP approximate speed and from Autonomie simulation

When the traffic pattern speed is available, the target speed is derived from it by adjusting it upward in order to account for the stops generated by the algorithm. Doing so ensures that the time needed to cover the sections is constant. For example, traffic pattern speed is available on the whole extent of the *Chicago2* and *Chicago3* trips; as a result, the ADAS RP trip time and simulation time are almost equal.

Table 3 – Simulated trips characteristics

Trip	Distance (km)	Avg. Speed (km/h)	Elevation Change (m)		Stops per km	Stop Time (%)
			Up (+)	Down (-)		
Boston	28.1	24.4	89.4	90.3	0.9	12.5
LA	14.8	25.6	0.0	0.0	2.8	60.2
SF	26.5	38.0	320.1	339.6	0.5	9.1
Chicago1	40.1	41.2	154.7	170.9	0.1	0.9
Chicago2	3.3	13.4	2.0	2.4	3.9	26.9
Chicago3	6.7	20.2	14.5	15.3	1.2	8.9
WashDC1	11.7	25.8	14.7	12.8	2.2	26.6
WashDC2	3.8	17.7	0.0	0.0	3.7	39.5
Detroit	21.3	36.3	45.2	44.3	0.8	16.5
Munich	35.7	54.5	174.1	91.5	0.3	6.1

On the other hand, when no traffic pattern data are available, the target speed is simply the raw ADAS RP speed – either the speed limit or “expected” speed. The assumptions are that the driving condition is close to free-flow and that the driver will target the speed limit outside of stop signs and traffic lights. Since the stop time can still be added, there may be a significant discrepancy between the “raw” estimated time and actual simulation time. In particular, this is the case for the LA trip, for which the traffic pattern speed was not available. Due to the numerous stop signs and traffic lights along the way, travel time almost doubles.

Table 3 compiles other quantities of interest for each trip.

### 5.3 Energy consumption prediction

One of the key applications of this newly developed process is to predict the energy consumption of vehicles driven along a real-world trip. To illustrate this capability, we chose to use the process to compare two vehicles with different powertrain architectures. The two vehicles evaluated on those cycles are:

- A conventional midsize car powered by an internal combustion engine (ICE) with a five-speed manual transmission and
- A 2004 Toyota Prius (a power-split HEV). The mass of both vehicles is set to 1400 kg. The electric accessory load is set to 600 W. Both vehicles are available in the public version of Autonomie. The Prius model was validated by using Argonne chassis dynamometer tests [17]. The control of the HEV is also the default one (no future road prediction). Fuel consumption reported for the HEV is “charge-balanced” (i.e., the net

battery energy is less than 1% of the fuel energy used).

Figure 13 shows the fuel consumption (F.C.) of both vehicles on the real-world trips defined previously. The cycles were sorted by ascending average speed. For comparison purposes, Figure 13 also shows F.C. on selected certification cycles, the European ECE and NEDC, as well as U.S. EPA urban (UDDS) and highway cycles (HWFET). The conventional car has a much higher F.C. than the HEV, ranging from almost 14 to 5.1 L/100 km; these seem like reasonable values for that type of car, and are comparable to F.C. on standard cycles. On the other hand, there is less variance in the HEV F.C., which stays between 2.8 L/100 km and 4.7 L/100 km.

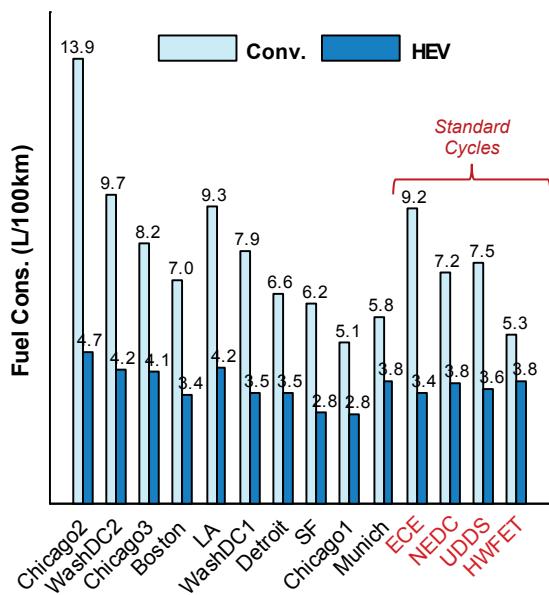


Figure 13: Fuel consumption of conventional vehicle and HEV; real-world trips sorted by ascending average speed

The F.C. for both vehicles tend to decrease with higher average speeds. This is true to real-world observations: as the average speed increases, the engine operates more efficiently, but at very high speeds, those gains are overcome by prevailing aerodynamic losses. The HEV gets the best F.C. reduction on trips where there are numerous opportunities to recuperate kinetic energy or with long idling periods. *Chicago2*, *WashDC2* and *LA* are routes in which there are a large number of stops (more than 2.8 stops per km) and stop time, (up to 60% for the *LA* trip). The conventional gets its worst F.C. and the HEV achieves the highest decrease in F.C. on those cycles. The HEV also gets a very low F.C. on *SF*, which

combines moderate speed highway driving with grades.

#### 5.4 Analysis of HEV operations in a hilly terrain

Grades often pose a different set of challenges for powertrain and controller designers, especially when it comes to hybrid vehicles. A long ascent is a challenge because the limited battery energy cannot overcome the engine power lost to engine downsizing. A descent, on the other hand, is beneficial, as regenerative braking can be used to recharge the battery. All standard cycles used for certification are “flat” (i.e., they do not include any grades). Real-world driving routes need to be used instead. One of the powerful features this new process offers is the prediction of grade for any trip on main roads. In this study, we can use that feature for a short analysis of hybrid operations.

The *SF* trip between San Francisco’s northern shore and San Rafael, 26 km to the north, in the northwest Bay area, includes several ascents and descents. It starts in urban San Francisco, crosses the Golden Gate Bridge, and turns into a three-lane highway. As shown in Figure 14, there is one major hill, between D = 2.2 km and D = 13 km and two smaller ones at 17.5 km and 22.5 km.

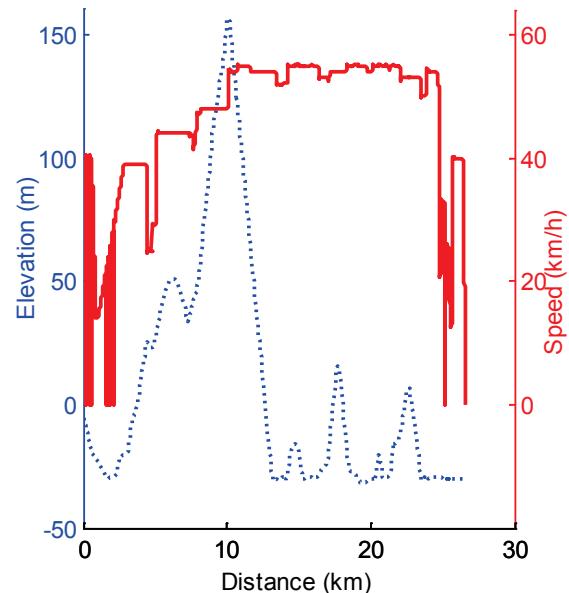


Figure 14: Elevation\* and speed on the SF trip in the San Francisco Bay area  
\*Reference: WGS84 Ellipsoid

The battery operations are shown in Figure 15 and Figure 16. In Figure 15, the battery  $\Delta$ SOC (which is the state-of-charge minus the initial SOC)

evolution is shown over the entire trip. The first part of the trip is in an urban environment and generally at low speed. That type of driving is often associated with electric-only mode. At  $t = 800$  s, a combination of positive grade and acceleration further depletes the battery. The long downhill grade that starts at  $t = 1200$  s is the opportunity to fully recharge the battery.

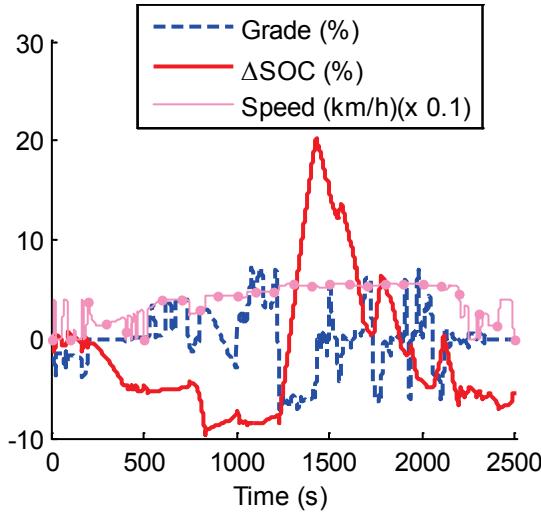


Figure 15: Battery SOC\* and grade on the SF trip  
\* $\Delta SOC = SOC - SOC(t = 0)$

As shown in Figure 16, the battery gets about 9 kW of electricity recharging it during the whole grade, leading the SOC to increase by almost 30 percentage points. That energy is then used throughout the cycle to displace fuel energy, leading to very good fuel consumption.

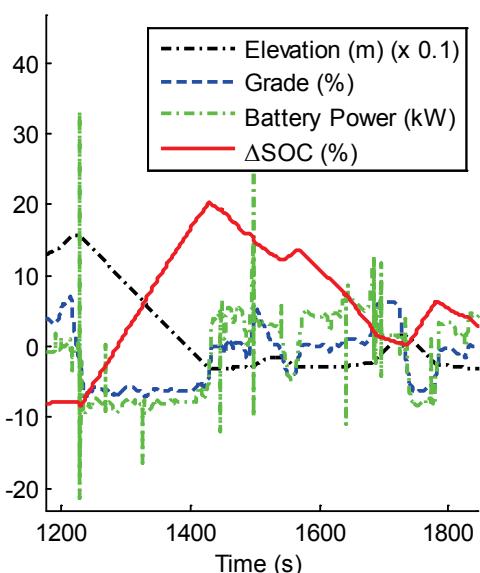


Figure 16: Battery power and grade on a hilly section of the SF trip

## 6 Conclusion

### 6.1 Future Work

Some key improvements to this method will be made in the future. More road information will be factored in the computation of the speed target (e.g., slower speed targets when a turning manoeuvre is identified).

Another major improvement will be the introduction of more variance to currently constant speed sections, in order to more accurately mimic stop-and-go traffic, natural speed oscillations, obstacles, etc. We will also emphasize validating the modeled speed profiles with real-world driving data.

### 6.2 Summary

A process was created to easily predict the energy consumption of a vehicle on a user-specified real-world trip. The user can define a trip by its starting and end points, in ADAS RP through a map interface. The saved trip is processed to create speed, stop, and grade profiles when it is loaded in Autonomie. The user-defined vehicle is then simulated over those profiles, and energy consumption is computed.

The processing algorithm is robust enough to deal with road links having varying levels of detail in the digital map. For example, the algorithm accounts for traffic pattern speed when it is available and on other information otherwise. The inclusion of stops in the target speed generation also creates realistic speed profiles, essential for estimating energy consumption.

We demonstrated that this process can be used not only to predict energy consumption but also to analyze the operation of any vehicle on a user-defined, real-world trip. We compared the fuel consumption of a Toyota Prius to that of a conventional car of the same class on 10 real-world trips and showed a significant benefit of the hybrid car. We also analyzed how the battery energy is managed on a trip with long grades.

This process opens new horizons, combining intelligent transportation systems (ITS) and transportation energy efficiency research. In particular, this tool is critical for intelligent trip-based control of electrified powertrains, battery EVs, HEVs, and plug-in HEVs. It provides a “proving ground” to test new smart controls. All the theories and methods of trip prediction can also be applied to an online, in-vehicle energy consumption horizon, which is indispensable for intelligent energy management.

## Acknowledgments

This work is funded by the U.S. Department of Energy Vehicle Technology Program. NAVTEQ graciously provided a free license of ADAS RP as well as support.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy 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.

## References

- [1] Advanced Driver Assistance Systems (ADAS): ADAS Research Platform, [www.nn4d.com/site/global/products\\_licensing/navteq\\_products/adas/50\\_tools\\_and\\_support/60\\_adas\\_rp/p\\_adas\\_research\\_platform.jsp](http://www.nn4d.com/site/global/products_licensing/navteq_products/adas/50_tools_and_support/60_adas_rp/p_adas_research_platform.jsp), accessed on Feb. 2012
- [2] Autonomie, [www.autonomie.net](http://www.autonomie.net), accessed on Feb. 2012
- [3] D. Karbowski, A. Rousseau, et al., *Plug-in Vehicle Control Strategy: From Global Optimization to Real Time Application*, 22th International Electric Vehicle Symposium (EVS22), Yokohama, October 2006
- [4] G.-E. Katsargyri, I. V. Kolmanovsky, et al., *Optimally Controlling Hybrid Electric Vehicles Using Path Forecasting*, 4613-4617, DOI 10.1109/ACC.2009.5160504, American Control Conference, 2009
- [5] G. Paganelli, S. Delprat, et al., *Equivalent Consumption Minimization Strategy for Parallel Hybrid Powertrains*, 55th IEEE Vehicular Technology Conference, 2076-2081, 2002
- [6] N. Kim, S. Cha, and H. Peng, *Optimal Control of Hybrid Electric Vehicles Based on Pontryagin's Minimum Principle*, IEEE Transactions on Control Systems Technology, 12791287, 2011
- [7] P. Sharer, A. Rousseau, et al., *Plug-in Hybrid Electric Vehicle Control Strategy: Comparison between EV and Charge-Depleting Options*, SAE paper 2008-01-0460, SAE World Congress, Detroit, April 2008
- [8] Y. Deguchi, K. Kuroda, et al., *HEV Charge/Discharge Control System Based on Navigation Information*, SAE Technical Paper 2004-21-0028, 2004
- [9] R. L. Walker, *Sentience – Using Electronic Horizon Data to Improve Hybrid Vehicle Fuel Economy*, Road Transport Information and Control, 2008
- [10] P. Sahlholm and K. Henrik Johansson, *Road Grade Estimation for Look-Ahead Vehicle Control Using Multiple Measurement Runs*, Control Engineering Practice, 1328–1341, Nov. 2010
- [11] D. Rotz, A. Bracht, K. Moran, et al., *Quality Assurance and Robustness for Predictive Cruise Control Using Digital Map Data*, SAE Technical Paper 2010-01-0467, 2010
- [12] R. Bartholomaeus, M. Klingner, et al., *Prediction of Power Demand for Hybrid Vehicles Operating in Fixed-Route Service*, World Congress IFAC, 5640–5645, 2008
- [13] J. Krumm, *Real Time Destination Prediction Based on Efficient Routes*, SAE Technical Paper 2006-01-0811, 2006
- [14] J. Froehlich and J. Krumm, *Route Prediction from Trip Observations*, SAE Technical Paper 2008-01-0201, 2008
- [15] Jurassic Test Green Race, [www.jurassictest.ch/GR](http://www.jurassictest.ch/GR), accessed on Feb. 2012
- [16] C. F. Minett, A. M. Salomons, et al., *Eco-routing: Comparing the Fuel Consumption of Different Routes between an Origin and Destination Using Field Test Speed Profiles and Synthetic Speed Profiles*, IEEE Forum on Integrated and Sustainable Transportation System, DOI 10.1109/FISTS.2011.5973621, 32–39, 2011
- [17] A. Rousseau, P. Sharer, et al., *Integrating Data, Performing Quality Assurance, and Validating the Vehicle Model for the 2004 Prius Using PSAT*, SAE Paper 2006-01-0667, SAE World Congress, Detroit, April 2006

## Authors



**Dominik Karbowski** holds a Master's degree in Science and Executive Engineering with a major in Energy Systems from Mines ParisTech, France. At Argonne, Dominik works on modeling advanced powertrains such as hybrids or plug-in hybrids, both light- and heavy-duty. He focuses his research on vehicle level control optimization and powertrain design optimization.



**Sylvain Pagerit** received a Master of Science in Industrial Engineering from the Ecole des Mines de Nantes, France, in 2000, and a Master of Science in Electrical Engineering from the Georgia Institute of Technology, Atlanta, in 2001. At Argonne, he focuses his work on the development of Autonomie and PSAT user interfaces, as well as automated optimization and sizing algorithms for advanced powertrains.



**Andrew Calkins** is a Lead Engineer at NAVTEQ's customer engineering services origination. He has more than 15 years of experience in developing location-aware software applications for customers. In the automotive and enterprise routing space, he is a recognized expert in the field, with frequent consulting requests to implement routing solutions that use the rich attribution set of the NAVTEQ map.