Chapter 9
Detailed Case Studies

In this chapter we present a few case studies that demonstrate and possibly expand the concepts presented earlier in this book in practical applications. Each case study is often a compilation of several research papers led by the authors and is meant to highlight system integration and practical considerations.

9.1 Eco-Approach to Signalized Intersections

In Sect. 1.3.2 an overview of published results on eco-approach to signalized intersections was presented. Section 4.1.3 provided a basic overview of signalized intersections and Sect. 7.7 presented more insight about potential energy saving via a fundamental numerical and analytical treatment.

In this section we provide a case study summarizing published results of a decade of work on the topic at Clemson University. More specifically deterministic and probabilistic planning of eco-approach to traffic lights [1, 2], impact on mixed traffic via traffic microsimulation analysis [3], and real-world experimental implementation [4] are discussed.

9.1.1 Numerical Approach

The goal is to find a velocity profile which reduces the energy consumption during a trip based on full or partial SPAT information. This problem can be formulated as an energy (fuel) minimization problem as was treated numerically in Sect. 7.7.1 and analytically in Sect. 7.7.2. Energy minimization requires inclusion of dynamic models of a specific vehicle and its propulsion system (ICEV, EV, HEV) to relate

A. Sciarretta and A. Vahidi, *Energy-Efficient Driving of Road Vehicles*,
Lecture Notes in Intelligent Transportation and Infrastructure,
https://doi.org/10.1007/978-3-030-24127-8_9

energy use to the velocity profile. To avoid the ensuing computational complexity and to decouple the choice of optimal speed from a vehicle's make and model, a simpler cost function can be used that penalizes a weighted sum of the total trip time and the acceleration and deceleration, instead of total energy use. The underlying assumptions in this choice are that idling at a traffic light and excessive acceleration that lead to braking both cost energy with little to no benefit to the driver. Other factors such as motion constraints imposed by red intervals, road speed limits, and the fact that very low velocities will be unacceptable to consumers, can be accounted for by constraining the solution space. In this case study, the fuel economy for a specific vehicle model is evaluated a-posteriori, by feeding the optimal velocity profile to a high-fidelity dynamic model of the vehicle.

We first describe, in Sect. 9.1.1.1, the scenario when deterministic and accurate SPAT information over the entire prediction horizon is available. When the phase and timing of upcoming signals are uncertain, a probabilistic term can be added to the cost function, as described in Sect. 9.1.1.2.

9.1.1.1 Planning with Deterministic SPAT Information

When only a single traffic light is on the horizon, one can use deterministic knowledge of green and red intervals to plan within the allowable velocity limits a timely arrival at a green as shown in Fig. 9.1.

To obtain a best achievable energy efficiency baseline, the optimal control problem is first solved assuming full and deterministic knowledge of signals' phase and timing over the planning horizon. Instead of the cost function in (8.2), a simpler and heuristic cost function is used that can be later transformed to a quadratic program for efficient numerical solution. The following cost function was chosen,

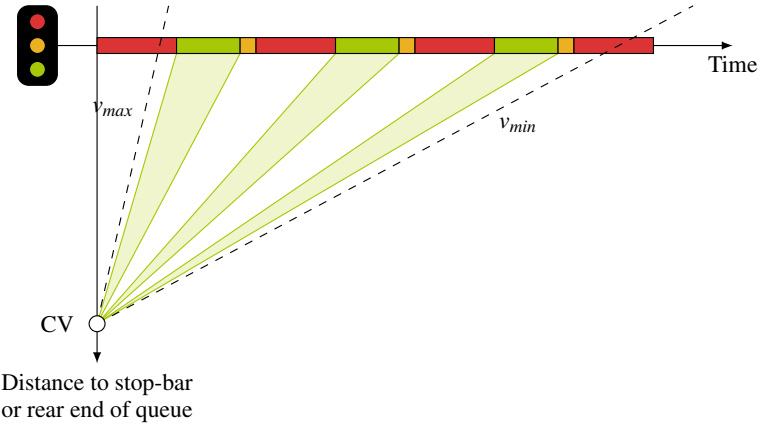

Fig. 9.1 Feasible velocity intervals in order to avoid stopping at red, if possible

$$J = \sum_{i=k}^{k+N-1} \left[w_1 \frac{\Delta t_i}{\Delta t_{min}} + w_2 \left| \frac{a_i}{a_{max}} \right| + c(s_i, t_i) \frac{1}{\varepsilon} \right] , \tag{9.1}$$

where J is the total cost and is indexed over position s with index i that starts at the current position step k and ends N steps later at step $k + N - 1$ with N being the length of the prediction horizon. Here $\Delta t_i = t_{i+1} - t_i$ is the time required for a vehicle to cover the fixed distance $\Delta s = s_{i+1} - s_i$ between steps s_i and s_{i+1} given the velocity at s_i and the acceleration a_i; Δt_{min} is the minimum time to complete the step if starting and ending at the maximum velocity and is used as a scaling factor, a_i is the constant acceleration assumed during step i, and a_{max} is the maximum allowed acceleration. The constants w_1 and w_2 are weighting terms. Motion constraints imposed by a red interval are imposed as a soft constraint by inclusion of the term $c(s_i, t_i)\frac{1}{\varepsilon}$ in the cost function. The value of $c(s_i, t_i)$ is zero except for spatiotemporal intervals when a light is red in which case its value is set to one, and ε is a very small constant (for example 10^{-6}), such that idling at red is discouraged.

The vehicle kinematics, realized by the following two-state equations, are imposed as equality constraints. Here s is the independent variable, velocity v and time t are the two states, and acceleration a is the input:

$$\frac{dv(s)}{ds} = \frac{a(s)}{v(s)} , \tag{9.2}$$

$$\frac{dt(s)}{ds} = \frac{1}{v(s)} , \tag{9.3}$$

Discretizing the above equations with a constant sampling interval of Δs and with a zero-order hold on acceleration, we obtain:

$$v_{i+1} = \sqrt{(v_i)^2 + 2a_i \Delta s} , \tag{9.4}$$

$$t_{i+1} = t_i + \frac{2\Delta s}{v_i + \sqrt{(v_i)^2 + 2a_i \Delta s}} , \tag{9.5}$$

The hard inequality constraints: $v_{min} \leq v_i \leq v_{max}$ and $a_{min} \leq a_i \leq a_{max}$ are also enforced. Here v_{min} and v_{max} are the road speed limits and can also include lowest speed acceptable to a driver; a_{min} and a_{max} are the feasible bounds for deceleration and acceleration.

The above optimal control problem can be solved numerically using Dynamic Programming (DP) and based on the discretization on position, time, and velocity as schematically shown in Fig. 9.2. The solution is calculated in one backwards sweep along the position axis taking advantage of Bellman's principal of optimality. The outline of the DP algorithm was described in Algorithm 1 of Sect. 6.2.2.2.

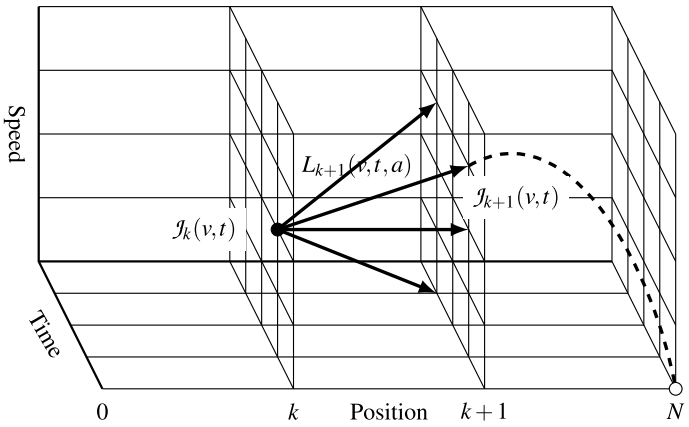

Fig. 9.2 Schematic of the DP grid

9.1.1.2 Planning with Probabilistic SPAT Information

Because perfect full-horizon SPAT information is not always available as explained in Sect. 8.2.4, here we consider the scenario where SPAT is known only probabilistically for instance based on historical data. The cost function in (9.1) is modified to the following to take into account the probabilistic nature of SPAT information,

$$J = \sum_{i=k}^{k+N-1} \left[w_1 \frac{\Delta t_i}{\Delta t_{min}} + w_2 \left| \frac{a_i}{a_{max}} \right| + c(s_i, t_i) |\ln\left(p(s_i, t_i)\right)| \right] . \tag{9.6}$$

All parameters and variables in (9.6) are the same as those described for (9.1); the only new variable is $p(s_i, t_i)$ which represents probability of green at time t_i for a light situated at position s_i. Therefore higher costs are assigned to solutions that pass through time intervals where probability of green is lower. At the limit when probability of green at s_i and t_i is zero, $\ln\left(p(s_i, t_i)\right) = \infty$ and passing through a red would be discarded. Where $p(s_i, t_i) = 1$, this term of the cost function drops to zero and increases the likelihood that the corresponding velocity will be selected. The probability of green for each light can be generated based on real-time and/or historical information as described in Sect. 8.2.4. Minimization of the cost function (9.6) with the equality and inequality constraints described in the previous subsection, remains a deterministic optimal control problem. The problem is solved using DP but in a receding horizon manner; as new information becomes available, the DP is re-solved taking into account the updated information over the remaining trip horizon. Note that an alternative approach will be a Stochastic Dynamic Programming (SDP) formulation where the expected value of the cost in (9.1) is penalized.

9.1.2 Simulation Results

9.1.2.1 Single Vehicle Simulation

A vehicle driving down a 800 m long street with three traffic signals was simulated in three scenarios. In the first scenario it is assumed that the vehicle has no information about the future state of the traffic light, in the second scenario real-time probabilistic SPAT information as described in Sect. 8.2.4 is assumed, and the third scenario full a-priori knowledge of SPAT information is assumed. Each scenario was run 1000 times in a Monte Carlo type experiment in which the start of red phases were randomized within a window of sufficient length for the vehicle to complete the route. The total cycle length, and length of each red were kept constant. Also the proportion of red to green times across all simulations were constrained to be the same. The start of each red of a traffic signal was chosen independently of the start of red of the next traffic signal. In all simulations, the penalty weights in the cost functions (9.1) and (9.1) are set to $w_1 = 1/8, w_2 = 1/8$. The value of ε is set at 10^{-6}. To solve the DP, the solution space is discretized to distances of 20 m, time increments of 1 s, and velocity steps of 1 m/s as schematically shown in Fig. 9.2. In this discretization grid choice, the computational time and memory requirements were reasonable for implementation on a PC.

In calculating the fuel economy, it was not computationally feasible to run all cases through a simulation cycle with a full high-fidelity vehicle model. Therefore a simplified vehicle model was developed using efficiency maps taken from AUTONOMIE [5] and a simplified gear shifting logic. For example, the effects of engine start and stop transients on fuel economy were not modeled in the simplified fuel economy calculations. The simulated vehicle is a two-wheel-drive, automatic transmission, conventional-engine vehicle. This vehicle had a total mass of 1580 kg, an engine producing a peak of 115 kW, and a constant electrical load of 200 W. The velocity profiles generated by the dynamic program were fed to this model to calculate the fuel economy for each case. This considerably simplified model provides a significant reduction in computational time when calculating the fuel economies for large numbers of simulation cases.

The Monte-Carlo simulation results found in Table 9.1 indicate that, for the road conditions described and with only real-time information and the probabilistic models, an average of 16% increase in fuel economy could be expected, representing approximately 62% of the benefit of full and exact traffic signal timing information.

With Monte-Carlo simulations indicating positive results for fixed time traffic lights, next an example of traffic signals with adaptive timing is presented. Twenty four hours of recorded traffic signal timing data from a series of signals in an urban corridor in Northern California is used. Figure 9.3 shows the lengths of the green phases for four different movements of one of these traffic signals.

A vehicle was simulated driving through the three traffic signals every 10 min over the 24 h yielding a total of 144 simulated drives per level of information. The real-world distance between the signals is preserved in the simulation, such that the

Table 9.1 Monte-Carlo simulation results reflect the positive influence of information, on average, on fuel economy

	Mean (MPG)	Standard deviation (MPG)
No information	25.9	5.0
Real time information	29.9	3.7
Full information	32.5	3.0

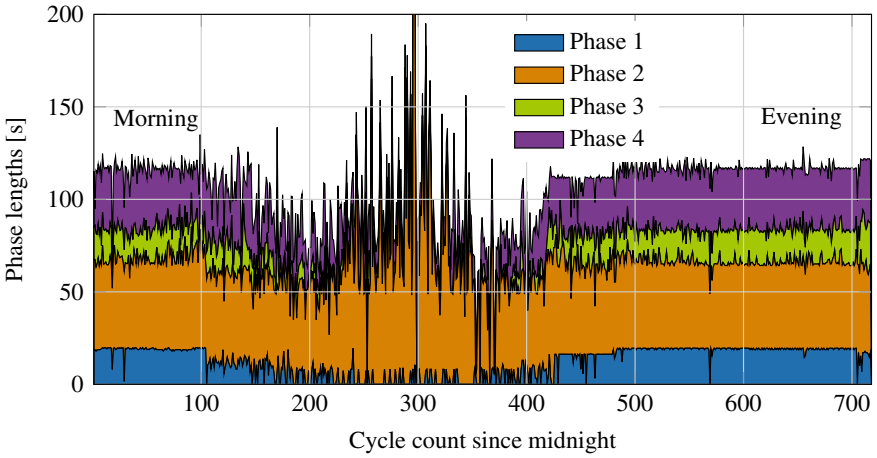

Fig. 9.3 Histories of green phases of four different movements of a traffic light on the chosen real-world route, for every cycle over a 24 h period (midnight to midnight)

Table 9.2 Fuel economy results from recorded real-world traffic signal timings with simulated vehicles moving between the lights reflect the positive influence of information

	Mean(MPG)	Standard deviation (MPG)
No information	31.7	3.1
Real time information	33.7	3.0
Full information	34.5	3.6

simulated vehicle has to cover the same distance using the same traffic signal timing offsets as a real driver would encounter. The total simulation distance is 1320 m. The lights occur at 520, 800, and 1200 m mark from the start. The DP resolution is the same as those set before. No other vehicles are considered to be on the road. In the case of real-time information the probability of green is calculated using Eq. (8.16) using a 24 h average of red and green lengths as t_r and t_g. If more relevant averages (for example a short-term average, a time of day average, or other statistical means) are available, they may continue to improve the performance of this real-time information case.

The simulation indicates that drivers with access to real-time probabilistic information were able to improve fuel economy over drivers with no information by approximately 6% (Table 9.2). This accounts for roughly 70% of the potential gains available through access to full and exact future knowledge of traffic signal timing. To deal with unexpected traffic, pedestrians crossing out of cross-walks, and other disturbances, the DP can be simulated frequently reproducing its cost-to-go map and optimal policy when necessary.

9.1.2.2 Multi Vehicle Microsimulation

While eco-approach to traffic signals could improve the energy efficiency of the ego vehicle, its impact on energy efficiency of upstream traffic deserves further investigation. One such microsimulation study was presented in [3]. Simulations were conducted in Quadstone Paramics [6] and custom code was developed to simulate vehicles with the eco-approach functionality. The equipped vehicles receive the timing of next upcoming traffic light in advance and adjust their speed for a timely arrival at green based on an analytical solution to the optimal control problem. The simulation is run in an urban corridor network. The 2.4 km (1.5 mile) path contains four signalized intersections with fixed timings. The speed limit of each link is 80 km/h (50 mph). Conventional vehicles do not have prior access to traffic signal information and always try to reach the maximum road speed limit unless affected by nearby vehicles or traffic signals. Three different traffic demand levels (300, 600, and 900 vehicles per hour per lane) and seven different penetration levels of equipped vehicles (100%, 90%, 70%, 50%, 30%, 10%, 0) are considered; therefore 21 simulations were conducted. Fuel consumption was estimated using a simplified model adopted from [7] which relates the fuel consumption rate to vehicle's velocity and acceleration. The parameters of the model are those in [7] and obtained by a third order polynomial fit to experimental data. It was further assumed that the engine was idling during negative acceleration consuming a constant idling fuel rate.

Figure 9.4 summarizes the energy consumption results for equipped vehicles as well as conventional vehicles. As shown in this figure, equipped vehicles (three bottom curves) consume much less fuel than conventional vehicles. This is due to fewer stops and closer-to-optimal operation of the engine. Another very interesting trend seen in Fig. 9.4 is that with the increment of the percentage of equipped vehicles, conventional vehicles consume less fuel. In other words, equipped vehicles have a positive impact on the energy efficiency of the entire mix of vehicles. With the increment of equipped vehicles, other conventional vehicles are more likely to follow them and benefit indirectly. However the energy efficiency of equipped vehicles generally decreases as their penetration increases. This could be due to slow-down of some equipped vehicles hindering procession of those behind them through a green.

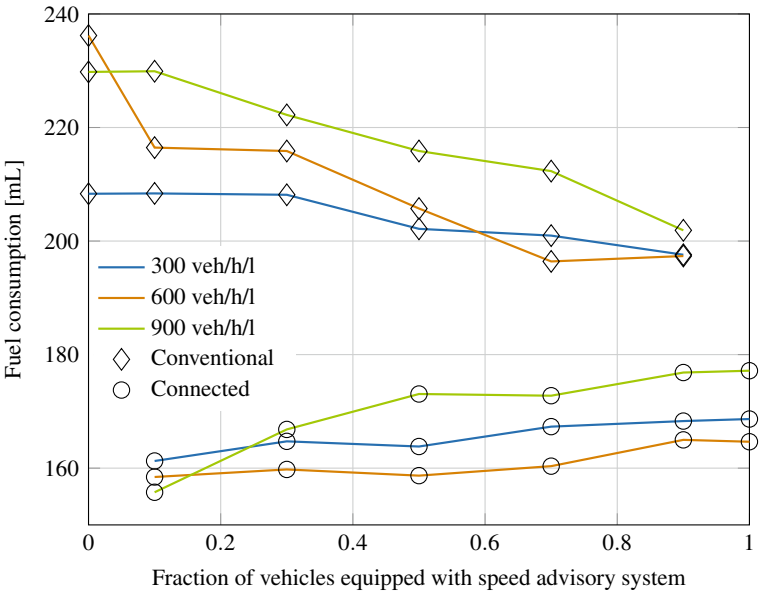

Fig. 9.4 Fuel consumption of vehicles with and without speed advisory system under different traffic demand levels and different penetration levels of equipped vehicles

9.1.2.3 Experimental Verification

Real-world implementation of traffic signal eco-approach or advisory has been reported in several recent publications. In [4] the concept is tested in the city of San Jose, California where the authors had real-time information of around 800 traffic lights. The connected vehicle identifies the next relevant traffic light and subscribes to it to receive updates about the state of the light using the cellular network via User Datagram Protocol (UDP) messaging. Because the vehicle only subscribed to the next upcoming light, a rule-based algorithm was employed to determine the feasible speed range for green arrival and use of dynamic programming was not needed. A user interface was created and recommended the appropriate speed range to pass through the next upcoming traffic signal during the green phase. The appropriate speed recommendation was displayed to the driver as green zones on the speedometer, as seen in Fig. 9.5.

A BMW 5 Series vehicle was used in these street experiments. Four drivers were asked to follow the speed recommendation shown on the dash display as long as safety was not jeopardized. The drivers were then asked to repeat the test, this time with the velocity advisory system deactivated. The tests were conducted in four different days and in real mixed traffic conditions. The fuel consumption of each driver was recorded for approximately one hour sessions and the results can be found in Table 9.3. In addition, the mean non-zero velocity, mean positive non-zero acceleration, and standard deviation of the same are all reported for evaluation of the

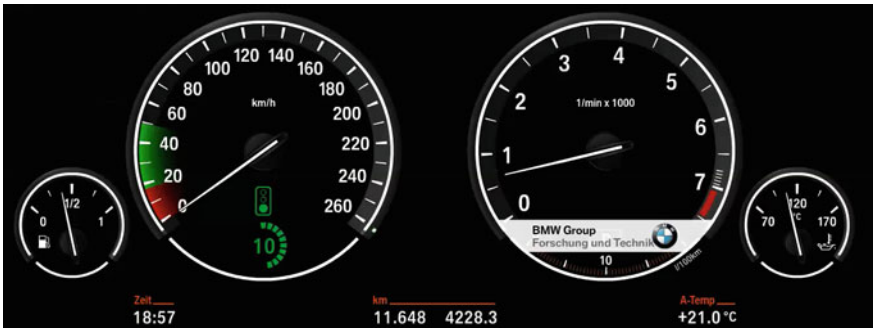

Fig. 9.5 Driver's dash display, including speed recommendation and countdown. A similar interface has been used in [8, 9]

Table 9.3 Field testing in San Jose, CA. Drivers were aware of the velocity advisory system and were specifically asked to follow the dash-display recommendations

Driver #	System inactive				System active			
	MPG	Mean velocity (m/s)	Mean accel. (m/s^2)	σ accel.	MPG	Mean velocity (m/s)	Mean accel. (m/s^2)	σ accel.
Driver 1	13.48	7.3	1.4	0.6	14.44	5.9	1.2	0.5
Driver 2	12.71	6.6	1.2	0.5	13.55	6.5	1.2	0.5
Driver 3	13.16	7.4	1.5	0.7	15.91	6.2	1.2	0.4
Driver 4	10.91	7.5	1.5	0.7	11.22	7.7	1.5	0.8

system. The fuel consumption measurements in Table 9.3 show that on average a 9.5% decrease in fuel usage is possible if drivers follow the displayed recommendations as closely as possible.

9.2 Cooperative Intersection Control

Results in previous section demonstrate that individual vehicles potentially save energy when they adjust their speed for a timely arrival at a green light. One could expect even higher efficiency with cooperative intersections in an all-autonomous vehicle environment as described in Sect. 1.4.3.

In recent years a few groups have proposed methods for such cooperative intersection control concepts. Here we summarize the results in [10, 11] in which the arrival time assignment was formulated as a Mixed Integer Linear Program and was implemented in a Vehicle-In-the-Loop (VIL) experiment.

9.2.1 Formulation as an Optimization Problem

Figure 9.6 schematically shows a shaded two-way intersection and the goal is to schedule the two conflicting directions of movements. This is done by defining a larger square centered at the intersection denoted by access area. Each vehicle i, is assigned an access time t_i, to the edge of this square when it is empty of vehicles from the opposing movement. The access area is sized large enough and based on the speed limit, to ensure there is enough time to react safely to a vehicle that violates its access time in the opposing movement.

The objective of increasing intersection throughput can be formalized here as an optimization problem. If n vehicles are subscribed at each instant to an intersection, minimizing the maximum assigned access times to these vehicles will push more vehicles through the intersection in a given time span. But the optimization objective could also consider the desired speed of each vehicle. One choice for the objective function can then be a weighted sum of both objectives:

$$J = w_1 \max(\{t_1, \ldots, t_n\}) + w_2 \sum_{i=1}^{n} |t_i - t_{des,i}| \,, \tag{9.7}$$

where t_i and $t_{des,i}$ are assigned and desired access times for vehicle i respectively and w_1 and w_2 are penalty weights. This optimization is expected to not only improve intersection flow but can reduce energy consumption due to reduced number of stops.

Given the speed limit v_{max} and the maximum acceleration constraint a_{max}, the earliest possible access time for vehicle i, denoted by $t_{min,i} = t_0 + \Delta t_1 + \Delta t_2$, can be calculated as illustrated in Fig. 9.7:

$$t_{min,i} = t_0 + \min\left(\frac{v_{max} - v_i}{a_i}, \frac{\sqrt{v_i^2 + 2a_i d_i} - v_i}{a_i}\right) + \max\left(\frac{d_i}{v_{max}} - \frac{v_{max}^2 - v_i^2}{2a_i v_{max}}, 0\right) , \tag{9.8}$$

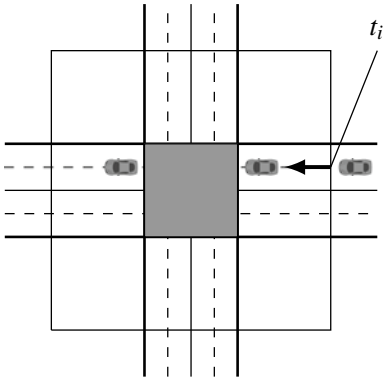

Fig. 9.6 Schematic of the proposed collaborative intersection control system. Gray denotes intersection area while white denotes access area

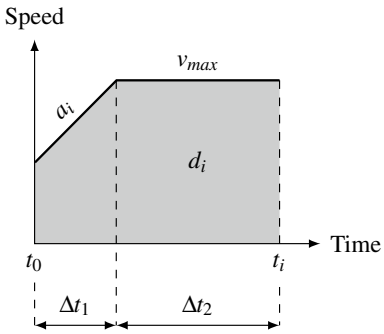

Fig. 9.7 The earliest possible access time based on speed limit and maximum accelerations

where t_0 is the current time, d_i is the distance of vehicle i to the intersection, v_i is the current speed of the vehicle i, and a_i is the maximum feasible acceleration for vehicle i thus yielding the minimum travel time. The minimum access time $t_{min,i}$ serves as a lower bound to the assigned access time for each vehicle:

$$t_i \geq t_{min,i} \ . \tag{9.9}$$

Two consecutive vehicles that are traveling on the same movement should be separated by a time headway of t_{gap_1} as they enter the intersection area. If vehicle j is the immediate follower of vehicle k in the same movement, the headway constraint can be expressed as

$$t_j - t_k \geq t_{gap_1} \ . \tag{9.10}$$

Two vehicles traveling on different phases (conflicting movements) need to be separated by a larger time headway to ensure that a vehicle can only enter the access area after all conflicting vehicles have left the intersection. For each two vehicles j and k that are on different phases of an intersection, the following OR constraint needs to be enforced:

$$t_j - t_k \geq t_{gap2} \quad \vee \quad t_k - t_j \geq t_{gap_2} \ , \tag{9.11}$$

where $\vee$ is the OR operator. The time headway between access times t_{gap_2} can be determined based on the dimensions of intersection and speed limits to allow enough time for a vehicle to come to a stop in the event that the vehicle in the opposing movement violates its assigned access time.

9.2.2 Numerical Solution

The optimization problem presented in Sect. 9.2.1 can be converted to a Mixed Integer Linear Program (MILP) using standard techniques. More specifically, in the cost

function (9.7) the term $\max(\{t_1, \ldots, t_n\})$ can be replaced by a new slack variable t_{max} and by imposing new constraints $t_i \leq t_{max}$. In the same cost function, the terms $|t_i - t_{des,i}|$ can be replaced by new slack variables δt_i and imposing $\delta t_i \geq \pm(t_i - t_{des,i})$. As a result the cost function (9.7) will be a linear function of the optimization variables t_i and the newly introduced slack variables. The constraints in (9.9) and (9.10) are also linear. Each disjunctive OR constraint in (9.11) can be converted to two linear constraints using the big M method and by introducing new binary variables $B_l \in \{0, 1\}$ and a large constant M as follows:

$$\begin{aligned} t_j - t_k + M B_l &\geq t_{gap_2} \ , \\ t_k - t_j + M(1 - B_l) &\geq t_{gap_2} \ , \end{aligned} \tag{9.12}$$

where $0 \leq l \leq m$, m is the number of disjunctive constraints, and j and k correspond to any two vehicles on different phases of an intersection. When $B_l = 0$, the first constraint in (9.12) indicates $t_j - t_k \geq t_{gap_2}$ and the second constraint in (9.12) is automatically satisfied given that M is sufficiently large. When $B_l = 1$ then $t_k - t_j \geq t_{gap_2}$ the first constraint in (9.12) is automatically satisfied and the second constraint is active.

The above optimization problem can then be written in a canonical linear programming form. That is to minimize $c^T x$ subject to $Ax \leq b$ and $x \geq 0$. Here $x = (t_1, \ldots, t_n, \delta t_1, \ldots, \delta t_n, t_{max}, B_1, \ldots, B_m)$ is the vector of optimizing variables, n is the number of all subscribed vehicles, and m is the number of artificial binary variables of Eq. (9.12). By including B_l for each disjunctive (OR) constraint among optimization variables, we ensure that the most favorable of the two OR constraints is chosen.

The above MILP problem was solved using IBM's CPLEX optimization package on an Intel Core i5@2.5 GHz Windows 7 laptop with 8 GB of RAM. For 50 subscribed vehicles, the average intersection controller execution time was 120 ms but varied between 28 ms and 2400 ms. These times include the MILP solver execution time plus the time needed for pre-processing the probe vehicle data and expressing the problem in canonical form. The MILP problem was solved once every 4 s to adapt to unmodeled effects and to deviation of vehicles from their assigned access times.

9.2.3 Simulation Results

Here we summarize some of the results detailed in [10, 12]. Figure 9.8 shows 9 vehicles approaching an intersection with X and O representing vehicles on conflicting movements. The distance of each to the intersection at the time of their subscription is shown on the vertical axis in the two plots of Fig. 9.9. The assigned access times are shown on top horizontal axis. Two extreme choices of penalty weights w_1 and w_2 are shown in the two plots. It can be seen that the MILP objective of reducing the

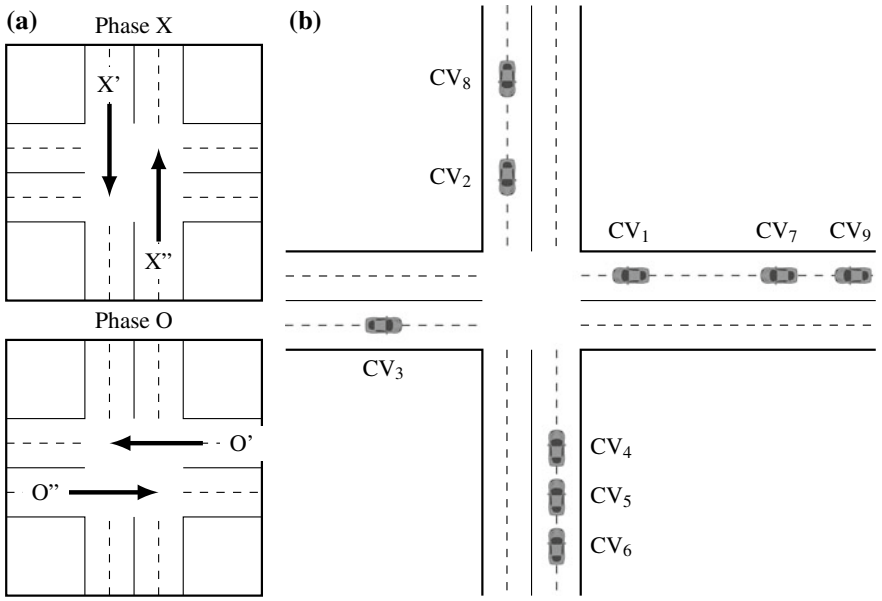

Fig. 9.8 Nine vehicles at a simplified two phase intersection with four movements

maximum assigned access time along with collision avoidance constraints groups together the vehicles of the same movement, when possible.

The performance of the proposed intersection control concept was also compared to a signalized intersection with a pre-timed traffic light in a one hour microsimulation with many vehicles. The intersection had four 500 m legs. The signal phase and timing for the benchmark pre-timed traffic signals were optimized off-line using Synchro signal optimization [13] resulting in a cycle time, green split, and yellow interval of 100, 44.5, and 3.5 s respectively. The vehicles were sampled from a negative exponential distribution [14] at 750 vehicles per hour for all four legs of the intersection. The vehicles' arrival pattern was recorded and replayed in all simulations. The average and maximum speeds were set to $v_{avg} = 15.6$ m/s (35 mph) and $v_{max} = 20$ m/s (45 mph). The pre-timed case was simulated twice: (i) with no speed advisory in which vehicles did not receive SPaT preview and (ii) with speed advisory to vehicles, similar to that of Sect. 9.1, in which vehicles received SPaT preview when they were within a 500 m range of the intersection. The penalty weights in the MILP objective function (9.7) were set to $w_1 = 50\%$ and $w_2 = 50\%$.

Table 9.4 summarizes some of the performance metrics for the three simulations that were conducted. It can be seen that number of stops has an almost 100-fold reduction with respect to a pre-timed signalized intersection. Average idle time for stopped vehicles is also cut in half. The average travel time shows considerable improvement as well. We expect the reduced idling and travel time positively impact energy efficiency. The experimental results presented next confirm this hypothesis.

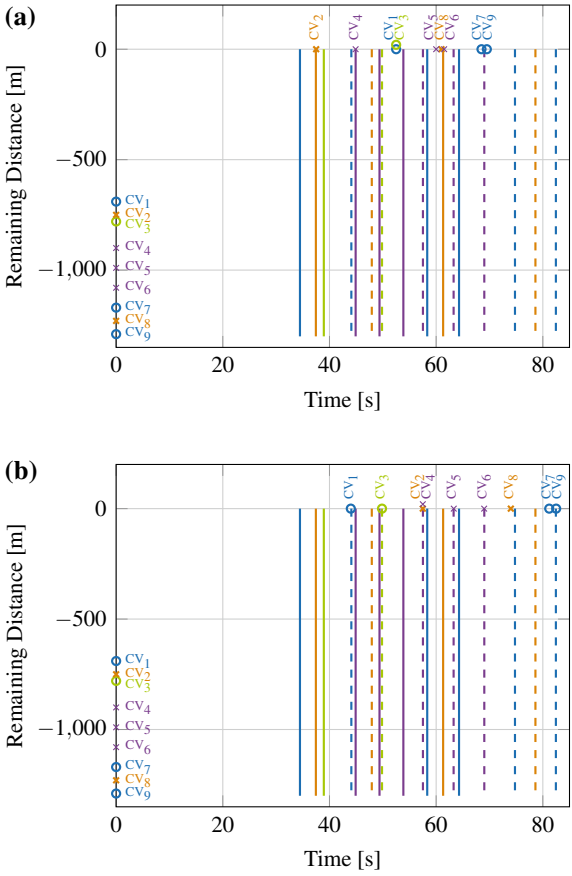

Fig. 9.9 Examples of MILP solution with 9 subscribed vehicles (vertical axis at time zero: remaining distance of each individual vehicle to the access area, horizontal axis at distance zero: access time assigned to each individual vehicle, solid lines: minimum access times $t_{min,i}$, dashed lines: desired access times $t_{des,i}$, $i \in [1, 9]$, colors and marks refer to the four directions of Fig. 9.8); **a** all weight given to intersection throughput improvement, **b** all weight given to satisfying the desired speeds of all vehicles [10]

Table 9.4 Microsimulation results comparing MILP controlled intersection concept with pre-timed signalized intersections with and without speed advisory

Performance metric	Pre-timed	Pre-timed+advisory	MILP
Intersection traversals	2900	2900	2900
Simulation time [min]	61	61	61
Number of intersection stops	1171	872	13
Total intersection idling Delay [min]	3640	1843	2
Avg. idle per stopped vehicle [s]	20	15	9
Avg. travel time per vehicle [s]	50	51	36

9.2.4 Experimental Results

In order to investigate the energy efficiency potential of the proposed approach, a Vehicle-In-the-Loop (VIL) test concept was proposed in [11] in which a real vehicle approaching a signal-less intersection on a test track interacts with hundreds of simulated vehicles approaching a simulated version of that intersection. Simulated and real vehicles all subscribe and communicate similarly to the intersection controller and are treated equally. The position of the real vehicle is injected in the microsimulation and therefore is easily visualized. The proposed approach is more realistic than a simulation-only environment, while also ensuring a safer environment for test vehicles because conflicting movements (and potential crashes) occur in a simulated environment. The VIL concept is shown in Fig. 9.10.

The test vehicle was a human driven Honda Accord LX with a 2.4 L 4-Cylinder SI gasoline engine. The vehicle was driven on an isolated straightaway located at International Transportation Innovation Center (ITIC) test track in Greenville, South Carolina. A custom-coded user interface on a mobile phone allowed speed control by the driver for a timely arrival at assigned access times. A cellular network was used for communication between the in-vehicle mobile phone and remote intersection controller. The mobile phone sent the vehicle position and velocity to the intersection controller every 5 s, received the assigned access time, calculated appropriate speed to meet this access time, and visualized the calculated speed by a narrow green arc on a circular speedometer. The traffic microsimulator node ran inside the test vehicle as shown in Fig. 9.10c for real-time monitoring of the simulations.

The fuel rate was estimated from data logged in real-time from the vehicle On-Board Diagnostics (OBD-II) port with details reported in [11]. Extra care was taken in calibrating the fuel rate estimator so that it matched the actual fuel consumption measured from a gasoline tank fill-up.

Within the same VIL framework described above 3 sets of tests were run: a pre-timed intersection baseline, pre-timed with speed advisory baseline similar to that of Sect. 9.1, and our proposed MILP controlled intersection. Each set consisted of 12 laps around the test track with wide U-turns at both ends of the track. The start time

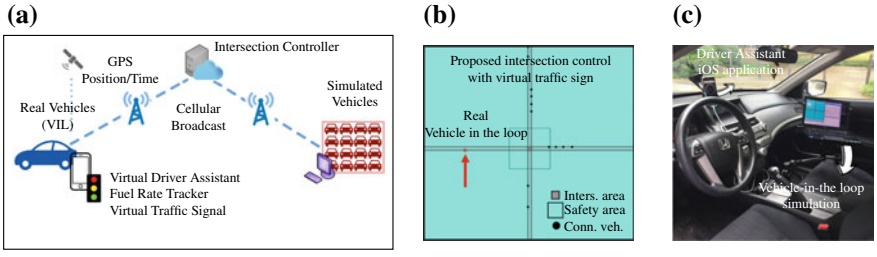

Fig. 9.10 Vehicle-in-Loop experimental setup in [11] showing **a** interactions between a real vehicle and microsimulation environment via 4G network, **b** Java microsimulation interface, and **c** in-vehicle setup

Table 9.5 VIL experiment results for real vehicle: Comparing MILP controlled intersection concept with pretimed baselines

Performance metric	Pre-timed	Pre-timed+advisory	MILP
Intersection traversals	12	12	12
Simulation time [min]	57.5	55	51
Number of intersection stops	10	0	0
Total intersection idling delay [min]	4.3	0	0
Avg. idle per stopped vehicle [s]	26	0	0
Avg. travel time per vehicle [s]	108	99	79
Fuel consumption [L]	1.13	1.11	0.91

at the beginning of each lap was randomized using a random number generator to prevent unintended bias due to cyclic runs.

Table 9.5 summarizes some of the performance metrics for the test vehicle. The test vehicle passed the MILP-based imaginary intersection 12 times without stopping. This resulted in 19.5 and 18.0% reduction in fuel consumption compared respectively with the two pretimed benchmarks.

9.3 Anticipative Car Following

In Sect. 1.3.3 we highlighted the potentials offered by CAVs for proactive and anticipative car following to lower energy use compared to current reactive car following practices. This approach is the eco-ACC introduced in Sect. 8.1.3. An analytic treatment of this scenario was presented in Sect. 7.8.

Here we expand on what was presented before in the book and present a detailed case study based on a compilation of the approach and results in [15–18]. We start by formulating car following as an optimal control problem. The unknown "disturbance" is the future position of the preceding vehicle which motivates methods for its deterministic or probabilistic prediction. We show via microsimulation analysis that one could gain on average by such anticipative car following measures.

9.3.1 Formulation as an Optimization Problem

In anticipative car following the goal is to reduce energy-consuming braking or stop and go events by judiciously adjusting the following distance between the two vehi-

cles as a buffer. This desire can be cast as an optimization with a cost function that balances the car following distance against acceleration command by penalizing a weighted sum of both as described in Sect. 8.1.3. For instance, in a model predictive approach the following quadratic cost minimization can be performed at the beginning of each receding horizon similar to that shown in [16],

$$\begin{aligned} \min_{u(i)} J &= w_s \|s_p(N) - s(N) - T\dot{s}(N) - L_{min}\|^2 + \\ &+ \sum_{i=0}^{N-1} \left(w_s \|s_p(i) - s(i) - T\dot{s}(i) - L_{min}\|^2 + w_u \|u(i)\|^2 \right) , \end{aligned} \tag{9.13}$$

where N is the number of time steps in a prediction horizon, progression of steps along the horizon is indexed by i, $u(i)$ are the acceleration commands and optimization variables, $\|.\|$ denotes the two norm and w_s and w_u are penalty weights. Here s_p and s are the position of the preceding and ego vehicles respectively, and L_{min} is the minimum desired gap between them when the ego vehicle is stopped. The distance headway $T\dot{s}$ is the product of time headway T and velocity of the ego vehicle $\dot{s}$ and is meant to induce larger gaps at higher ego vehicle speeds.

The vehicle longitudinal kinematics along with a first order lag between the acceleration command input u and the vehicle's acceleration a can be enforced as equality constraints and obtained by discretizing the following continuous time equations:

$$\dot{s}(t) = v(t) , \tag{9.14}$$

$$\dot{v}(t) = a(t) , \tag{9.15}$$

$$\dot{a}(t) = -\frac{1}{\tau} a(t) + \frac{1}{\tau} u(t) , \tag{9.16}$$

where τ approximates the time constant from acceleration command to actual acceleration. After discretization, the continuous-time t is indexed by i as the independent variable.

Hard constraints on vehicles states and on the following distance must be enforced at each step in time. An important safety constraint is a lower bound on car following distance. An upper bound can also be optionally enforced to avoid leaving large gaps that could negatively impact traffic flow or encourage cut-ins. In summary:

$$L_{min} \leq s_p(i) - s(i) - T\dot{s}(i) \leq L_{max} \quad i = 1, \ldots, N . \tag{9.17}$$

Minimum and maximum speed limits should also be enforced,

$$v_{min} \leq v(i) \leq v_{max} \quad i = 1, \ldots, N . \tag{9.18}$$

As shown in [17] and illustrated in Fig. 9.11, the powertrains maximum acceleration capacity depends strongly on velocity as seen in combined engine-transmission maps. The velocity dependent acceleration constraint can be approximated as piece-

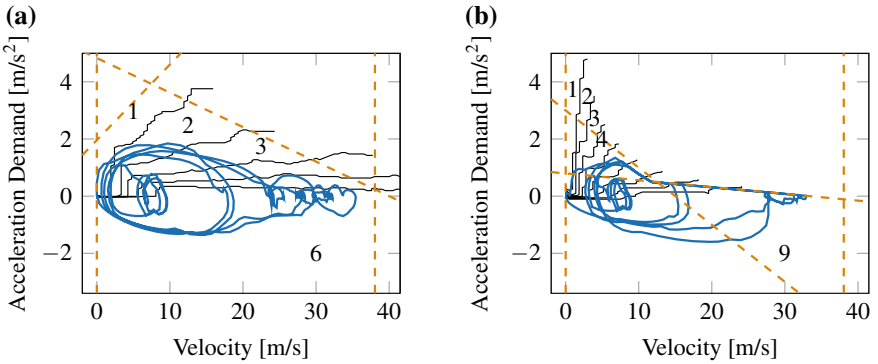

Fig. 9.11 Velocity dependent acceleration constraints (dashed orange) for **a** a passenger vehicle where conjunctive maximum acceleration constraints yield a convex velocity-acceleration admissible set and **b** a heavy duty vehicle where disjunctive maximum acceleration constraints yield a non-convex velocity-acceleration admissible set. In both scenarios, the blue phase portrait trajectories are sample operating point trace of vehicles running MPC planning under US06 drive cycle [17]

wise linear combinations of velocity and acceleration as detailed in [17] and illustrated in Fig. 9.11. Depending upon the convexity of the acceleration-velocity constraint-admissible set, these piecewise linear constraints may be applied conjunctively or disjunctively. As shown in [17] disjunctive OR constraints can be converted to conjunctive AND constraints by introducing new integer optimization variables in the big M method described in Sect. 9.2.2.

In [18] a terminal constraint is also imposed on velocity and position of the ego vehicle to prevent a collision post-prediction horizon. This terminal constraint on velocity and position of ego vehicle is constructed using kinematic relationships and assuming that the preceding vehicle will apply maximal braking post-prediction horizon (worst case scenario). While the resulting terminal constraint is nonlinear, a linear approximation to it could be used.

The above linear constraints along with the quadratic cost function in (9.13) form a quadratic program over each horizon. Efficient QP solvers exist that could solve this problem in real-time. Even when integer variables are introduced to handle disjunctive linear constraints shown in Fig. 9.11b, the resulting Mixed Integer Quadratic Program (MIQP) can still be solved relatively fast as documented in [17].

The main challenge is the uncertainty about the position of the preceding vehicle s_p over the optimization horizon. Note that s_p appears in the cost function (9.13) as well as in the constraint (9.17). The optimization could be solved under the worst case scenario assuming the preceding vehicle comes to a sudden emergency stop at each step of the horizon. Such worst case assumptions could induce very conservative and perhaps unnecessarily large headways between vehicles. Here we employ the methods described in Sect. 8.2.3 for predicting the motion of preceding vehicle.

With a probability distribution for $s_p(i)$, the gap constraints can be enforced probabilistically as a so-called chance constraint. For instance a minimum gap constraint in Eq. (9.17) can be instead written as:

$$\mathrm{P}(s(i) + T\dot{s}(i) \leq s_p(i) - L_{min}) \leq 1 - \alpha \quad i = 1, \ldots, N \ , \tag{9.19}$$

which means that the chance of violating the constraint should be less than $1 - \alpha$. Note that at any current step $i = 0$, s, $\dot{s}$, and s_p are all deterministic rather than probabilistic, and MPC finds solutions that do not violate constraints. The probabilistic constraint will be converted to a deterministic constraint using probability distribution of $s_p(i)$. If we denote $R_{1-\alpha}$ as the position where the cumulative distribution function of $s_p(i)$ is equal to $1 - \alpha$, then the equivalent deterministic constraint is

$$s(i) + T\dot{s}(i) \leq R_{1-\alpha} - L_{min} \ . \tag{9.20}$$

Similarly, the maximum distance constraint can be enforced probabilistically. With transformation of the probabilistic constraints to deterministic ones, we end up with a standard MPC problem. This is the approach employed in [15, 16].

9.3.2 Numerical Solution

In [18] a parameter optimization was performed to find optimal values of the penalty weights and the prediction horizon length. Once the parameters were fixed, the Gurobi optimization package [19] was used to solve the QP MPC problem for a passenger vehicle. For a heavy truck the maximum acceleration constraint is disjunctive as was illustrated in Fig. 9.11b which resulted in a Mixed Integer Quadratic Program (MIQP) formulation for the MPC problem. For both cases, passenger cars and heavy truck, two scenarios were considered: (i) When a CAV follows another CAV in which future intentions of the preceding CAV were available to the following CAV for the duration of prediction horizon, and (ii) when a CAV follows a conventional vehicle in which a probabilistic model similar to those described in Sect. 8.2.3 was used to estimate the intentions of the preceding vehicle.

Table 9.6 shows the computation times for both QP and MIQP MPCs. One MPC vehicle was simulated following an open-loop vehicle. The optimization was solved on a laptop PC equipped with 16.0 GB RAM and a 2.70 GHz CPU. In Table 9.6, Optimization Time refers to the time required to solve the mathematical program (QP or MIQP) and Compt. Time refers to the total time required to run a single vehicle's control move determination, including both preview handling and optimization.

Table 9.6 Computation time for anticipative car following MPC. The MPC is converted to a QP for a passenger car and to a MIQP for a heavy truck. When a CAV follows another CAV the preview source is connectivity and full intentions over the horizon is communicated from the preceding CAV. On the other hand when the preceding vehicle is a conventional vehicle, preview source is a probabilistic model

Algorithm	Preview source	Worst case in-horizon constraints	Worst case terminal constraints	Mean comp. time [s]	Max comp. time [s]	Mean opt. time [s]	Max opt. time [s]
Car	Connectivity	No	Yes	0.0337	0.0561	0.0108	0.0444
Car	Probabilistic	Yes	Yes	0.0757	0.1134	0.0110	0.0892
Truck	Connectivity	No	Yes	0.0435	0.0789	0.0148	0.0504
Truck	Probabilistic	Yes	Yes	0.0571	0.0919	0.0069	0.0425

9.3.3 Simulation Results

Here we report the results of the implementation in [18] in which a mixed string of 8 vehicles following a target vehicle are simulated. The mix includes conventional cars with no connectivity that use a standard Intelligent Driver Model (IDM) for car following as described in Sect. 4.2.1.3. The IDM parameters were sampled from empirical data. More specifically, the desired time headway and maximum and minimum acceleration levels were sampled from log-normal distributions fit to the empirical data of [20]. Different penetrations of CAVs in the mix are explored. Each CAV solves a variant of the receding horizon car following approach described above. When a CAV is immediately preceded by another CAV it receives the intended position of the preceding vehicle but when it is following an IDM vehicle it uses a probabilistic prediction of the preceding car motion. Post horizon, collision avoidance is ensured by considering worst case hard constraints at the end of each prediction horizon when two CAVs follow each other. When a CAV follows an IDM vehicle, worst case collision constraints are enforced along the prediction horizon as well.

Both passenger vehicle and heavy duty CAVs are injected in the mix. Observing that the penetration of heavy trucks is 25% in some US highways, 0, 1, or 2 heavy duty trucks are injected in the mix of the 8 vehicles in the string. The receding horizon problem for the passenger vehicle is converted to a QP and that of a heavy duty CAV becomes a MIQP as described in Sect. 9.3.1. A quasi-random approach is used to create different placement of the vehicle types in the string leading to a total of 2224 scenarios and simulations as detailed in [17]. Figure 9.12 shows the cumulative fuel consumption results at different CAV penetrations and for 0, 1, and 2 heavy duty vehicles. The fuel consumption was estimated using quasi-static engine fuel maps. As expected the energy efficiency increases with the penetration of CAVs. A comparison between homogeneous human-like IDM strings and those composed

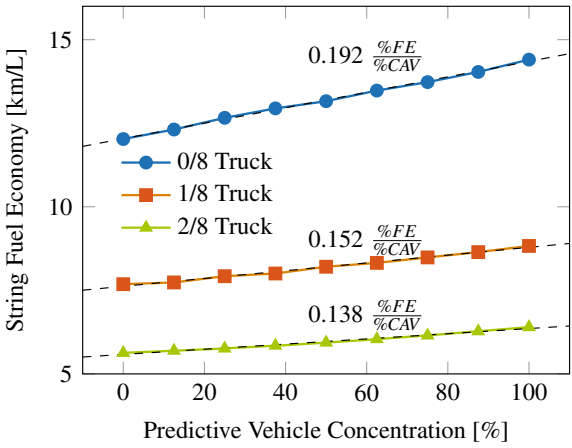

Fig. 9.12 Combined string fuel economy at various penetration levels of predictive and heavy vehicles

entirely of CAVs is available from the endpoints of each line in the plot. In absence of heavy duty vehicles a 1.9% improvement in fuel economy is shown for every 10% increase in CAV penetration.

9.4 Anticipative Lane Selection

In Chap. 4 mandatory and discretionary lane change behavior of human drivers were briefly discussed and the MOBIL lane change model was presented. Connected and automated vehicles can more judiciously make a lane change decision and execute it as explained in Sect. 1.3.4. Here we present a case study based on results in [21] in which lane selection is formulated as an optimization problem and in anticipation of neighboring vehicles' intentions.

9.4.1 Formulation as an Optimization Problem

A lane command u_l as well as a longitudinal acceleration command u_a are the high level control inputs. The longitudinal state space Eq. (9.16) relates u_a to vehicle longitudinal acceleration, velocity, and position. The lateral position of the vehicle $l(i)$ is expressed in lane width units with respect to a reference frame that aligns integer values of l with lane centers. For example, on a two-lane road $l = 1$ could coincide with the center of the right lane, $l = 2$ denotes the center of the left lane, and $l = 1.5$ is on the visible marking between the lanes. In [21] a critically damped second order lag is assumed between the lane command u_l and the actual lane l of the form

$$\ddot{l}(t) + 2\zeta\omega_n\dot{l}(t) + \omega_n^2 l(t) = \omega_n^2 u_l(t) \; . \tag{9.21}$$

The damping ratio ζ is chosen to be unity and the natural frequency ω_n is chosen to be 1.1 rad/s resulting in a settling time of around four seconds to match a naturalistic lane change behavior. Binary "lane indicator" variables will be defined later that determine whether each lane is occupied as a function of l.

The state dynamics in (9.16) and (9.21) are discretized, transforming the continuous-time formulation with t as the independent variable into a discrete-time problem with i as the independent variable. The cost function (9.13) can be modified to include lane choices along with new constraints that arise in multi-lane traffic. Similar to the approach in Sect. 9.3.1, the following moving horizon cost function can be used,

$$\min_{u_a(i),u_l(i)} J = \sum_{i=0}^{N-1} \left(w_v(v(i) - v_{ref}(i))^2 + w_a(u_a(i))^2 + w_l(l(i) - l_{ref}(i))^2\right) + w_v(v(N) - v_{ref}(N))^2 + w_l(l(N) - l_{ref}(N))^2 , \tag{9.22}$$

where $i = 0$ denotes the current time step and N is the prediction horizon. Here v_{ref} is the desired velocity and l_{ref} denotes the desired lane[1] that could be dictated from the vehicle navigation system. For instance a vehicle may prefer to stay in the right-most lane in anticipation of an imminent exit. The above cost function strikes a trade off between remaining in the desired lane and following the desired velocity while it penalizes the acceleration command to reduce unnecessary braking and energy loss. Minimizing the moving horizon cost could command changing to a faster lane to pass a slow moving vehicle instead of reducing speed in the desired lane. After passing the slow vehicle, return to the desired lane could be commanded due to the residual lane cost. This behavior is consistent with what a human driver would do.

The position of the ego vehicle s in its lane is constrained by other reachable vehicles in the same lane. The ego can be either behind or in front of another vehicle which is a non-convex OR constraint:

$$s \leqslant s_{rear,p} - L_{ego} \quad \vee \quad s \geqslant s_{front,p} , \tag{9.23}$$

where $\vee$ is the OR operator. Here $s_{rear,p}$ denotes the position of the rear bumper of the neighboring vehicle p and $s_{front,p}$ is the position of its front bumper. The length of the ego vehicle is denoted by L_{ego}. Similar to the approach in Sect. 9.2 the disjunctive OR constraint can be converted to an AND constraint using the big M method:

$$s \leqslant s_{rear,p} - L_{ego} + M\beta_p \quad \wedge \quad s \geqslant s_{front,p} - M(1 - \beta_p) , \tag{9.24}$$

where $\wedge$ is the AND operator. Here $\beta_p \in \{0, 1\}$ is a new binary variable defined for each reachable obstacle p in the ego vehicle lane and M is a large enough

[1] l_{ref} does not need to be an integer, in fact it is useful to offset it slightly to break symmetry in common situations.

constant. When $\beta_p = 0$, the first constraint in (9.24) may be active and indicates $s \leqslant s_{rear,p} - L_{ego}$ and the second constraint in (9.24) is trivially satisfied given that M is sufficiently large. When $\beta_p = 1$ then $s \geqslant s_{front,p}$ and the first constraint in (9.24) is automatically satisfied. The optimal solution chooses the value for β_p that results in minimum value of the cost function in (9.22).

Finally for each lane a new "lane indicator" binary variable $\mu_{nl} \in \{0, 1\}$ can be defined. When the ego is fully or partly in lane nl, $\mu_{nl} = 1$, otherwise if it does not occupy lane nl at all $\mu_{nl} = 0$. The inequality constraints in (9.24) are only relevant if the two vehicles occupy the same lane. So in multi-lane scenarios the constraint in (9.24) should be expressed as

$$s \leqslant s_{rear,p} - L_{ego} + M\beta_p + M(1 - \mu_{nl}) \quad \wedge \quad s \geqslant s_{front,p} - M(1 - \beta_p) - M(1 - \mu_{nl}) \,, \tag{9.25}$$

which should be imposed for every reachable vehicle in the occupied (or to be occupied) lanes.

The lane indicator binary variables μ_{nl} are determined as a function of ego vehicle lane state $l(i)$. For instance in a 2-lane scenario only two of these binary variables μ_1 and μ_2 are defined. Consider the variable δ to be the maximum deviation, in units of lane width, for a vehicle to remain wholly in a lane. If $1 + \delta \leq l(i) \leq 2 - \delta$ then the vehicle occupies both lanes and we set $\mu_1 = \mu_2 = 1$. If $l(i) \leq 1 + \delta$ then vehicle is in lane 1 and we set $\mu_1 = 1$ and $\mu_2 = 0$. If $l(i) \geq 2 - \delta$ then vehicle is in lane 2 and we set $\mu_1 = 0$ and $\mu_2 = 1$.

Setting these lane indicators must be handled by using inequality constraints as proposed in [21] since the MIQP solver demands a certain canonical form that does not accommodate if-then-else rules. For instance in a two lane scenario the following four constraints correctly set the values for μ_1 and μ_2,

$$-l(i) - M\mu_1 \leq -(2 - \delta) \,, \tag{9.26}$$

$$l(i) + M\mu_1 \leq M + 2 - \delta \,, \tag{9.27}$$

$$l(i) - M\mu_2 \leq 1 + \delta \,, \tag{9.28}$$

$$-l(i) + M\mu_2 \leq -1 - \delta + M \,, \tag{9.29}$$

where again M is a large enough number and $\mu_1, \mu_2 \in \{0, 1\}$ are binary optimization variables to be determined for each time step.

9.4.2 Numerical Solution

The above moving horizon optimization was solved using the Gurobi optimization package [19]. In order to reduce the computational effort of a mixed integer quadratic program which can be high for reasonable choices of prediction horizon, a move blocking approach is used in [21] to reduce the number of integer variables. The lane

Table 9.7 Simulation results of anticipativelane selection algorithm

	Time [min]	Fuel [L]
Rule-based	13.8	17.1
MPC	12.9	15.7
Free flow	12.7	15.3

change command u_l is held constant over every three steps while the acceleration command u_a can assume a different value at each step of the prediction horizon. With a sampling time of 0.4 s, a 10 s prediction horizon could be handled in real-time. With move blocking, longer horizons could be executed real time all on a laptop PC equipped with a 4 core, 2.7GHz CPU and 16GB RAM. More details can be found in [21].

9.4.3 Simulation Results

In [21] the simulated receding horizon approach was tested first in a two lane scenario in MATLAB with 4 MPC CAVs in their desired first lane encountering a slow moving vehicle. Each CAV has a different desired speed and a set of full factorial simulations was performed to include all possible placements of the CAVs with respect to each other. The MPC algorithm was compared to a reactive rule-based algorithm in which the CAVs used an IDM model for car following. They changed lane reactively when slowed down by a user defined threshold behind the slow moving vehicle provided that necessary space in the adjacent lane was available.

As shown in Table 9.7, the MPC algorithm reduced fuel consumption by 8.4% and travel time by 6.2% compared to the rule-based algorithm in full factorial simulations. The results were also compared to free flow traffic results. Excess travel time, defined as the increase in travel time over the congestion-free value, decreased by 79% relative to the reactive algorithm. Correspondingly, excess fuel consumption was reduced by 80% compared to a baseline of 18.1 mL for the average vehicle.

9.5 Eco-Routing and Eco-Coaching

Eco-routing has been extensively described in Chap. 5. Eco-coaching, an implementation variant of eco-driving is the subject of Sect. 8.1.1. This section is mainly based on the publication [22], which collects various results of the European Commission funded project OPTEMUS (2015–2019).[2] The scope is to discuss the energy efficiency benefits of the eco-routing and eco-coaching functions as experimentally assessed on a demonstrator car.

[2] www.optemus.eu.

9.5.1 Experimental Setup

In order to evaluate the impact of the eco-routing and eco-coaching (Sect. 8.1.1) strategies in real-world conditions, an experimental setup as illustrated in Fig. 9.13 was implemented. The system consists of:

- A dedicated smartphone application, which serves as a Human-Machine Interface (HMI) and hosts the eco-driving algorithms ensuring high response time;
- A cloud computing server which communicates with a Geographical Information System (GIS) to retrieve real-time traffic data and hosts the routing and driving range algorithms which are both costly in terms of computation time;
- An On Board Diagnosis (OBD) dongle which monitors the battery state-of-charge. This device is optional and could be replaced by an observer.

For the pre-trip and in-trip assistance, the driver can obtain from the HMI the most energy-efficient route, as well as the energy driving range available with the current battery state-of-charge. Furthermore, to perform in-trip eco-driving assistance, the smartphone application computes the optimal speed on the previous sub-trip at each

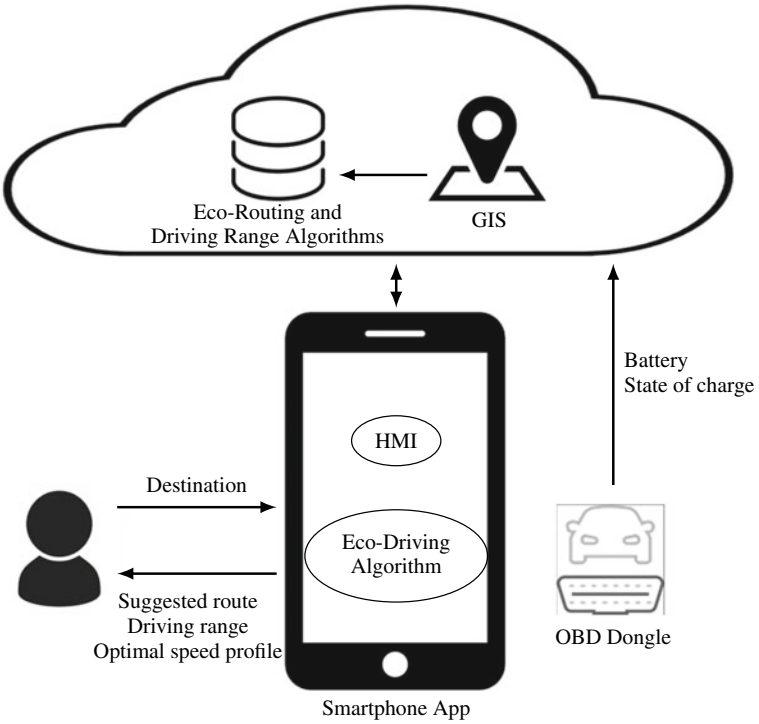

Fig. 9.13 System architecture for experimental validation of eco-routing and eco-coaching algorithms

speed breakpoint, as shown in Fig. 8.2. The application shows the optimal trajectory together with the actual speed profile in order to provide the drivers with visual feedback on their driving style. An energy consumption evaluation of the driving style on the sub-trip is also shown.

9.5.2 Experimental Results

In order to experimentally assess the energy benefits in using the proposed strategies, a series of field tests has been conducted. The tests have been conducted in the urban and sub-urban area of Turin, Italy, with a Fiat 500e (83 kW, 200 N electric motor) driven by a professional driver.

9.5.2.1 Energy Consumption Model

For the energy consumption model validation, a total of 35 trips were recorded, featuring an overall traveled distance of 434.6 km and a total travel time of 16.1 h (i.e. an average of about 12.4 km and 27.5 min per trip). The vehicle reference data for model validation were provided by the CAN-Bus data acquisition system. The topological data (i.e. road network, road signs, etc.) and the traffic information (i.e. average speeds) were provided by HERE [23].

The presented models for energy consumption and travel time, in Eqs. (4.41) and (4.31) respectively, were compared to standard state-of-the-art approaches. In particular, the energy consumption model was compared to a simple model reducing Eq. (4.28) to the average speed V_i for the entire road link ("NOacc"), and yet a simpler version neglecting also the auxiliary power term ("NOacc+NOaux"). Analogously, the travel time model was compared to a simple model obtained by considering only average velocity, $\tau_i = \ell_i / V_i$ in (4.31) ("NOacc"). The results of the experimental validation in Table 9.8 are expressed in terms of symmetric Mean Absolute Percentage Error [24] (sMAPE[3]) with respect to the CAN-bus measured energy consumption and travel time. The energy consumption estimation performed by the different models for one specific case is shown in Fig. 9.14. The proposed model largely outperforms the state-of-the-art approaches. Accuracy of the prediction models is crucial for a reliable navigation strategy.

[3] sMAPE is an accuracy measure based on percentage errors. It is used to solve the issue of heavier penalty on negative errors than on positive errors.

Table 9.8 Experimental validation results of the energy consumption model

	Energy			Time	
	Proposed model	NOacc	NOacc+ NOaux	Proposed model	NOacc
sMAPE	8.5%	17.4%	30.4%	7.9%	13.4%

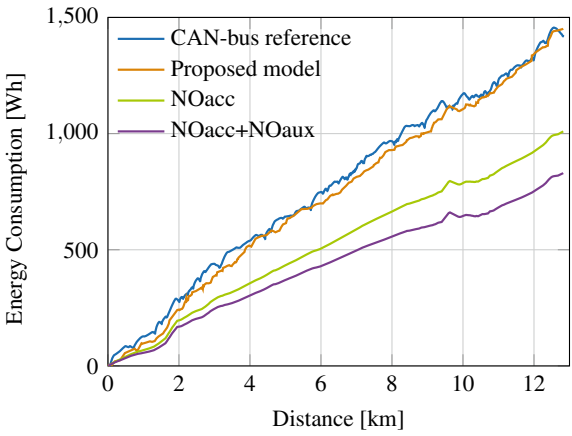

Fig. 9.14 Energy consumption estimations over one trip compared to the CAN-bus reference measurement

9.5.2.2 Eco-Routing

The aim of the conducted experiments for eco-routing validation is threefold. The first goal, achieved in simulation, is to show that a bi-objective eco-routing is highly effective in discarding those energy-optimal routes that penalize travel time. This feature could increase drivers' compliance to the route planning assistance. The second goal, achieved in simulation, is to prove that the shortest route is likely to be out of the Pareto efficient solutions in terms of energy consumption and travel time. The third goal, achieved both in simulation and experimentally for a selected origin/destination (O/D) pair, is to show that the eco-route is actually more energy-efficient than the shortest and the fastest route.

The energy and time weights w_k (5.7) and τ_k (5.15) were computed by using average traffic speed information for a regular working day at 09:00. The non-dominated points in the objective space calculated by the proposed algorithm are shown in Fig. 9.15. The eco-route and the fastest route correspond to the two solutions for $\lambda = 1$ and $\lambda = 0$ in formulation (5.17). A route corresponding to one of the Pareto-optimal trade-offs is labeled as multi-route. The shortest route is away from the Pareto front of the non-dominated points, and therefore not interesting either in terms of energy consumption nor in terms of travel time. The four routes are displayed on a map in Fig. 9.16.

For the experimental routing validation, the professional driver was instructed to drive on the eco, the shortest, and the fastest routes previously identified in simulation.

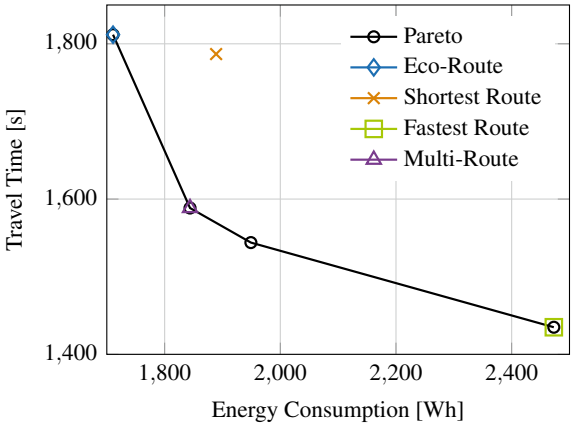

Fig. 9.15 Routing simulation results. Non-dominated points in the criteria space calculated by the proposed algorithm. The performance of the four routes is also displayed

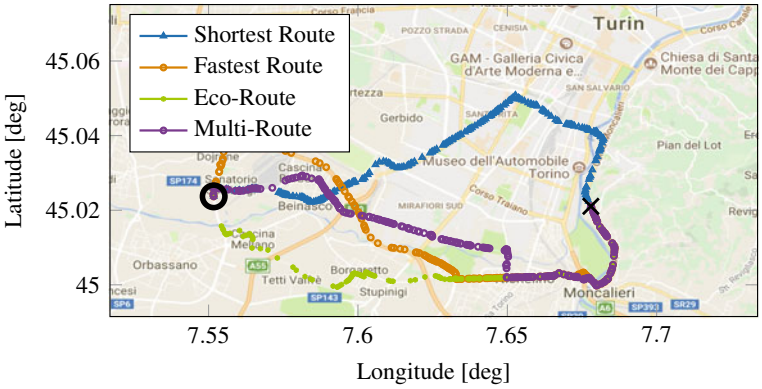

Fig. 9.16 The four routes obtained in simulation for the selected origin/destination pair

The driver performed three repetitions for each route, starting the experiments always at the same hour of the day, therefore over several working days. As summarized in Table 9.9, the experimental results showed that the eco-route for the identified O/D pair is actually the most energy-efficient among the three alternatives. In particular, the eco-route shows on average an energy gain of 4.5% with respect to the shortest route, and 12.4% with respect to the fastest one. In terms of energy prediction accuracy over the three repetitions of each route, the sMAPE (between the measurement and the estimation) ranged from 4.5% for the eco-route to 9.3% for the fastest route. In terms of travel time prediction, the sMAPE ranged from 3.5% for the eco-route to 12.7% for the fastest.

Table 9.9 Experimental eco-routing results

		Energy			Travel time	
		Mean	sMAPE	Gain	Mean	sMAPE
Eco	CAN	1784	4.5%	/	1637	3.5%
	Prediction	1759.5			1696	
Short	CAN	1868	5.9%	4.5%	2236	10.6%
	Prediction	1854			2230	
Fast	CAN	2041	9.3%	12.4%	1457	12.7%
	Prediction	2305			1655	

9.5.2.3 Driving Range Estimation

As discussed in Sect. 5.2 typical strategies for the estimation of electric vehicles driving range make assumptions on the average energy consumption per kilometer. Such an average energy consumption, often corresponding to the worst-case consumption for conservative estimation, is then used to calculate the driving range in terms of distance.

In Fig. 9.17, the proposed strategy for the calculation of the driving range is compared to a typical approach based on an average energy consumption and the corresponding driving range in terms of distance. In the experiment, which can be conducted only in simulation, a conservative average energy consumption of 0.18 kWh/km was chosen (the value is consistent with a worst-case energy consumption observed during the experimental campaign). The available energy capacity was set to 1 kWh. The chosen average energy consumption translates into a radius of 5.5 km, which corresponds to a quite symmetric driving range as shown in blue in Fig. 9.17. However this approach neglects important factors such as road grade, traffic conditions, type of employed route. The proposed strategy is able to take into account all these aspects, and every destination inside the driving range may be reached by following an eco-route. The driving range (in green in Fig. 9.17) is asymmetric about the origin due to the presence of hilly terrain in the road network and different consumption patterns. In this example, the energy driving range varies from a minimum of about 5 km to a maximum of about 10 km.

Furthermore, it may happen that the region is not simply connected, as discussed in [25], meaning that some destinations in the driving range are unreachable with the current battery state of charge, even by following an eco-route. Such critical destinations may be shown to the driver for more precise assistance, and they are shown with orange dots in Fig. 9.17. In this case, the unreachable destinations correspond either to points close to the driving range boundary or to particularly energy-expensive roads, such as motorways.

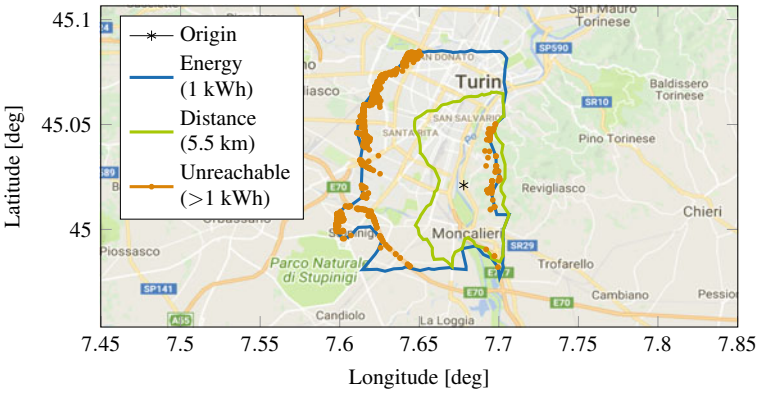

Fig. 9.17 Comparison of the energy driving range based on the prediction of energy consumption and the standard distance driving range

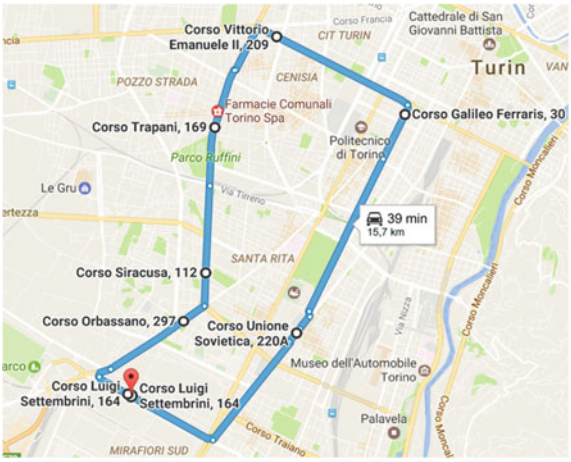

Fig. 9.18 Itinerary for the eco-coaching experimental campaign. The itinerary is about 16 km long, with an estimated travel time of 40 min

9.5.2.4 Eco-Coaching

The eco-coaching strategy was tested in the city center of Turin. The following test procedure was followed: (i) three repetitions of the itinerary given in Fig. 9.18 without eco-driving assistance (i.e. "pre-eco"), (ii) the driver was then introduced to the eco-driving smartphone application, and (iii) three repetitions of the itinerary were then performed with the eco-driving assistance (i.e. "eco").

An example of the corresponding speed profiles is given in Fig. 9.19. Table 9.10 shows the energy consumption measured from the CAN bus with and without the eco-driving assistance. It shows that with the eco-coaching assistance the energy consumption was reduced by 9% on average, while travel time was reduced by 3%. It is therefore possible to improve the energy efficiency of the trips without driving

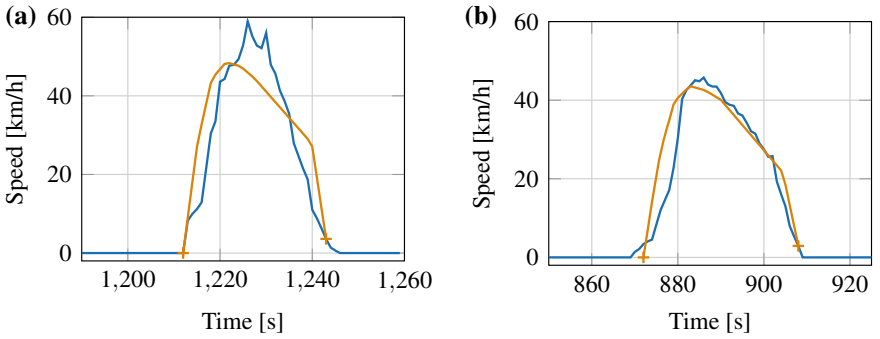

Fig. 9.19 Example of measured (blue) and optimal (orange) vehicle speed profiles for the same road segment without (**a**) and with the eco-coaching assistant (**b**)

Table 9.10 Experimental eco-coaching results

	Energy [Wh]	Time [s]	Speed[km/h]
Average of pre-eco trips	2246	2623	19.9
Average of eco trips	2041	2543	21.3
Variation	−9%	−3%	+3%

more slowly since the average speed is not decreased. One additional fact is that *each* eco-trip made with the eco-driving assistance has a lower energy consumption.

9.5.2.5 Overall Gains

Eco-routing and eco-coaching have been validated independently on a similar urban driving conditions with the same vehicle and driver. On average, the eco-coaching and eco-routing allow the driving range to increase by 9 and 12% respectively (as compared to the fastest route), by reducing the energy consumption. The driving range prediction strategy allows the driver to have a more precise knowledge of which destination is reachable. The state-of-the-art iso-distance approach is significantly less precise and is therefore necessarily tuned in a conservative way. By overcoming this limitation, the proposed strategy allows the driver to use the full potential of the available driving range.

References

1. Asadi Behrang, Vahidi Ardalan (2011) Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. IEEE Trans Control Syst Technol 19(3):707–714
2. Mahler G, Vahidi A (2014) An optimal velocity-planning scheme for vehicle energy efficiency through probabilistic prediction of traffic-signal timing. IEEE Trans Intell Transp Syst 15(6):2516–2523
3. Wan N, Vahidi A, Luckow A (2016) Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic. Transp Res Part C: Emerg Technol 69:548–563
4. Mahler G, Winckler A, Fayazi SA, Vahidi A, Filusch M (2017) Cellular communication of traffic signal state to connected vehicles for eco-driving on arterial roads: system architecture and experimental results. In: Proceedings of intelligent transportation systems conference, pp 1–6. IEEE
5. Argonne National Lab. Autonomie vehicle system simulation tool (2019). https://www.autonomie.net/
6. Paramics Q (2009) The paramics manuals, version 6.6. 1. Quastone Paramics LTD, Edinburgh, Scotland, UK
7. Kamal MAS, Mukai M, Murata J, Kawabe T (2011) Ecological driving based on preceding vehicle prediction using MPC. IFAC Proc Vol 44(1):3843–3848
8. Xia H, Boriboonsomsin K, Schweizer F, Winckler A, Zhou K, Zhang W-B, Barth M (2012) Field operational testing of eco-approach technology at a fixed-time signalized intersection. In: Proceedings of international conference on intelligent transportation systems (ITSC), pp 188–193. IEEE
9. Andreas W, Andreas W (2013) Advanced traffic signal control algorithms, appendix A: Exploratory advanced research project. Technical report, BMW final report
10. Fayazi SA, Vahidi A (2018) Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing. IEEE Trans Intell Veh 3(3):287–299
11. Fayazi SA, Vahidi A (2017) Vehicle-In-the-Loop (VIL) verification of a smart city intersection control scheme for autonomous vehicles. In: Proceedings of conference on control technology and applications (CCTA), pp 1575–1580. IEEE
12. Fayazi SA, Vahidi A, Luckow A (2017) Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP. In: Proceedings of American control conference (ACC), pp 4920–4925. IEEE
13. Trafficware. Synchro studio (2019). https://www.trafficware.com/synchro.html
14. Mathew TV (2014) Transportation systems engineering. Cell Transmission Models, IIT Bombay
15. Zhang C, Vahidi A (2011) Predictive cruise control with probabilistic constraints for eco driving. In: Proceedings of dynamic systems and control conference and symposium on fluid power and motion control, pp. 233–238. American Society of Mechanical Engineers
16. Wan N, Zhang C, Vahidi A (2019) Probabilistic anticipation and control in autonomous car following. IEEE Trans Control Syst Technol 27:30–38
17. Dollar RA, Vahidi A (2017) Quantifying the impact of limited information and control robustness on connected automated platoons. In: Proceedings of international conference on intelligent transportation systems (ITSC), pp 1–7. IEEE
18. Dollar RA, Vahidi A (2018) Efficient and collision-free anticipative cruise control in randomly mixed strings. IEEE Trans Intell Veh 3:439–452
19. Gurobi Optimization. Gurobi optimizer 5.0 (2013). http://www.gurobi.com
20. Pourabdollah M, Bjärkvik E, Fürer F, Lindenberg B, Burgdorf K (2017). Calibration and evaluation of car following models using real-world driving data. In: Proceedings of international conference on intelligent transportation systems (ITSC), pp 1–6. IEEE
21. Dollar RA, Vahidi A (2018) Predictively coordinated vehicle acceleration and lane selection using mixed integer programming. In: Proceedings of dynamic systems and control conference, pp V001T09A006–V001T09A006. American Society of Mechanical Engineers

22. De Nunzio G, Sciarretta A, Gharbia IB, Ojeda LL (2018) A constrained eco-routing strategy for hybrid electric vehicles based on semi-analytical energy management. In: Proceedings of international conference on intelligent transportation systems (ITSC), pp 355–361. IEEE
23. HERE. HERE APIs (2019). https://developer.here.com/develop/rest-apis
24. Armstrong JS (1985) Long-range forecasting. Wiley, New York ETC
25. De Nunzio G, Thibault L (2017) Energy-optimal driving range prediction for electric vehicles. In: Proceedings of intelligent vehicles symposium (IV), pp 1608–1613. IEEE