FUTURE MOBILITY: INTEGRATING VEHICLE

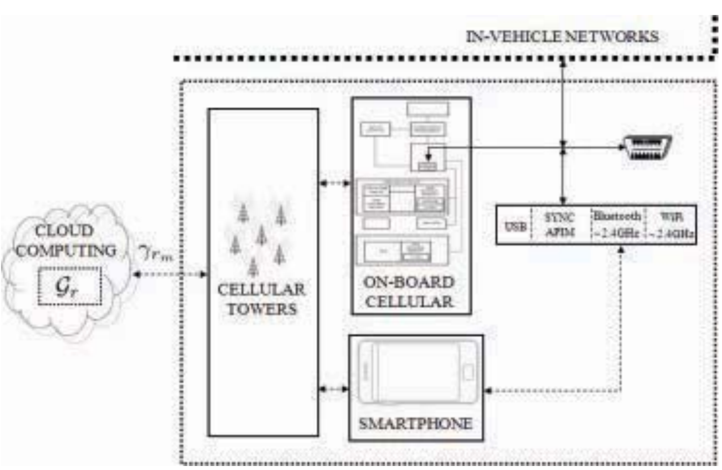
This article presents an automotive control approach for information-rich future mobility. It integrates in-vehicle networked controls with cloud computing accessible through a wireless-network to elevate current on-board controls to a new level for additional benefits and performance. While in-vehicle controls remain essential for safety critical and real-time functionality, the cloud-computing paradigm offers another degree of freedom for control system design.

AUTOMOBILE CUSTOMERS AROUND THE WORLD ARE DEMANDING NEW TECHNOLOGIES FOR FUNCTIONS THAT GO BEYOND TRADITIONAL NEEDS. Automobile manufacturers also offer new features for brand recognition and for meeting regulatory requirements. While superior mechanical design remains critical for meeting technological challenges, additional effort is being focused on enhancing electronics, controls, and software[1]. Automotive manufacturers are investing in the smart utilization of information. We believe that cloud computing offers new opportunities for maximizing the benefit of using "big data" generated by modern automobiles. Cloud computing enables network access to a shared pool of configurable computing resources which have virtually "unlimited" storage space and computational power. It can be rapidly provisioned and released with minimal management[2, 3]. Its presence in automotive applications has been rather limited so far[4, 5]. For example, a cloud-computing service in MyFord Mobile uses an onboard wireless module to communicate with a cloud-computing service for a whole array of infotainment and telematics features[6]. Progressive Insurance was the pioneer in using cloud services for monitoring driver performance. Its MyRate driving-monitoring device transmits driving data to a cloud server to determine the driver's insurance premium.

Cloud computing is useful in optimizing controls for adaptive driving experiences. The current on-board Electronic Control Units (ECUs) are targeted for about 100 distinct control and diagnostic functions. While automotive companies continue upgrading ECUs to meet increased computational needs, augmenting them with flexible computing resources presents a feasible alternative. By combining in-vehicle networks and cloud-computing resources, ECUs can conduct simpler and safety critical (e.g., traditional) computations in real-time while more complex but less time-critical computations are accomplished via cloud computing.

Outsourcing computation-intensive tasks to a cloud-computing server is an extension of the current server-based concierge/infotainment type features, e.g., GM's OnStar service and Ford's SYNC service. Unlike safety and time critical tasks, higher-level computations (e.g., route planning, optimizing speed profile, context dependent control calibrations, model updating, and diagnosis) might be conducted remotely and used locally. Furthermore, intelligent agents can be called in the cloud to optimally guide vehicles for fuel economy, ride comfort, safe driving, etc. Such intelligent agents can also initialize the computation with a good initial guess for optimization (e.g., using historical or community data), thereby permitting fast convergence to the optimal solution. Various optimization needs in ECUs that were not feasible before can now be conducted in the cloud through a local-simple-remote-complex strategy.

FIG. 1 A CCS connected to in-vehicle networks via communications hardware.



INTEGRATING IN-VEHICLE NETWORKS WITH THE CLOUD-COMPUTING SERVER

We will refer to the cloud-computing resources as the Cloud Computing Server (CCS). Integrating the CCS with the in-vehicle networks needs a new system consisting of cloud enabled hardware, communication device, services provided by the CCS, software agents, etc., as shown in **Fig. 1**.

Let's use MyFord Mobile for electrified vehicles as an example to describe the hardware requirements[6]. The MyFord Mobile app and website enhance the electric vehicle experience. They help manage the charging process and provide notification if the planned trip is within the vehicle's range. It includes an onboard wireless module, which can communicate with both remote cellular towers and in-vehicle networks. The driver can also use a smartphone to access the same CCS or the in-vehicle networks. The cellular module consists of a backup battery, a power supply management system, a control area network (CAN) controller, a

CONTROL WITH CLOUD COMPUTING

BY DIMITAR FILEV, JIANBO LU AND DAVOR HROVAT

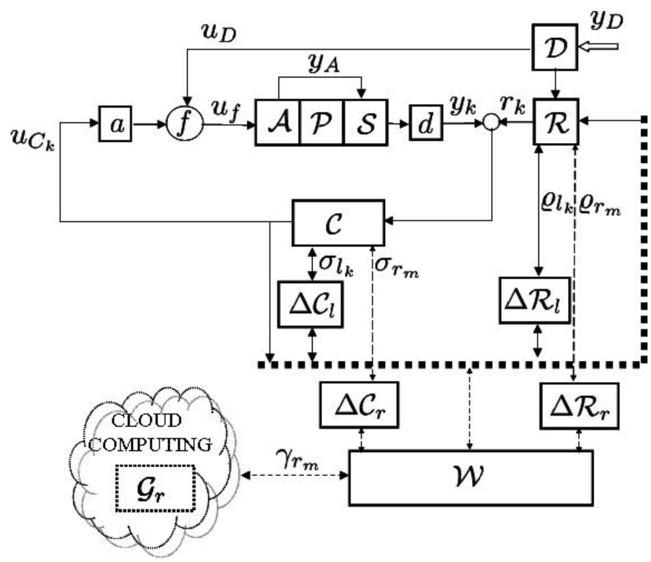


FIG. 2 Diagram of in-vehicle networks augmented with the remote cloud computations – modules and signals: $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{C}$ -collections of all vehicle actuators, sensors, and controllers; $\mathcal{P}$ - a vehicle as a generic plant; $\mathcal{D}$ - a driver, $\mathcal{R}$ - reference set-points; $\Delta\mathcal{C}_l$ and $\Delta\mathcal{R}_l$ - incremental controls and reference set-points operating based on local vehicle measurements; $\Delta\mathcal{C}_r$ and $\Delta\mathcal{C}_r$ - incremental controls and set-points operating based on both local and cloud information; $\mathcal{W}$ - wireless communication device; *a* ,*d* - A/D and D/A converters; (f) - the collection of all the arbitration operations among different control signals; u_{C_k} - digital control signal generated from the controller $\mathcal{C}$; u_D - driver's control commands that are influenced by unknown information y_D observed by the driver; y_A - sensor measurements at the actuators, y_k digital sensor measurements; r_k - reference signal.

subscriber identity module (SIM) card, a 2.5G Hz global system for mobile (GSM) communications circuit block, and a global positioning system (GPS) antenna (optional). Several functions have been implemented, including: estimating state of charge, programming vehicle charging, using location-based services, receiving alerts if the vehicle isn't charging or its charging status has changed, remotely locking or unlocking doors, downloading personalized information, etc.

To actively reduce the wireless load and storage usage, the communication between the in-vehicle networks and the CCS cannot be conducted continuously using bi-directional data flow with the same high sampling rate as in-vehicle networks. It should be performed in a controlled manner to reflect the on-demand nature of the application and to assure communication integrity. Precautions are required in integrating remote data with local data. Typically remote data are not used in highly dynamic and safety-critical applications, but are used for slowly time-varying processes, where the delay effect is negligible. For example, road slope changes relatively slow and a several second time delay in remote data will not cause significant error in road slope related computations.

The CCS entrusts remote services with a user's data, software, and computation. Its most relevant cloud-computing architecture is Software as a Service (SaaS) where vehicles can access the software in CCS and the vehicle's electronics does not need to manage the cloud infrastructure on which the application software is running. While the in-vehicle software has a standard architecture[7], its integration with the cloud computing software over the wireless network is still an open question[8]. An agent-based architecture for networked control systems is proposed in[9] that can be applied to integrating in-vehicle networks with the CCS. The agent-based approach is appealing in that the algorithms and software associated with the agents can be run on-demand and called by many users as needed, while in ECUs, the software is fixed. The agent-based approach leads to a clear shift from previous proprietary systems to modular systems[3].

The other relevant architecture includes storage as a service where the states related to the vehicle, the driver, and the driving conditions are all categorized, summarized, and stored under the appropriate labels in the CCS. They can be accessed by the host vehicle anywhere and anytime. They can also be accessed by the CCS's supervisory agents to assemble community states that summarize the historical and current states of multiple vehicles.

Fig. 2 shows a sketch of the integration. $\boldsymbol{W}$ is the communication device, connecting in-vehicle networks and the CCS. The collection of data storage, data processing, computing, and software units in the CCS is denoted as $\boldsymbol{G}_r$ which generates remote signals that can be packed and sent to $\Delta\boldsymbol{C}_r$ and $\Delta\boldsymbol{R}_r$. Similarly, data processed in $\Delta\boldsymbol{C}_l$ and $\Delta\boldsymbol{R}_l$ can be sent to $\boldsymbol{G}_r$. The architecture of the integrated system involves the complexity of in-vehicle networks and $\boldsymbol{G}_r$, and can be summarized as follows:

- **local-simple-remote-simple:** simple computations in ECUs and $\boldsymbol{G}_r$ data utilization for enhancing local computations
- **local-simple-remote-complex (LSRC):** simple computations in ECUs and complex computations in $\boldsymbol{G}_r$
- **local-complex-remote-simple:** complex computations compatible with ECUs' capability and $\boldsymbol{G}_r$ data utilization for enhancing local computations
- **local-complex-remote-complex:** complex computations compatible with ECUs' capability and even-more resource-demanding computations in $\boldsymbol{G}_r$

We envision three main types of cloud-computing agents: state estimation, supervisory control, and crowdsourcing agents, for automotive applications. The *state estimation agents* extend the ability of the on-board system to estimate variables that are not directly measured. The *supervisory control agents* execute broad tasks that require intensive computational resources, e.g.

learning models of driver and vehicle, conducting constrained optimization, calibrating control algorithms in real-time, etc. The *crowdsourcing agents* gather and summarize data from multiple vehicles, and fuse the data with other web enabled information sources.

STATE ESTIMATION AGENTS

In this section, we address estimating states/conditions using cloud computing agents including state estimation (cloud as sensor), vehicle health monitoring (cloud as storage), driver's state in ECUs and generating an evolving driver model in G_r (cloud as computer and storage).

Cloud Based State Estimator

State estimation from limited sensor measurements is usually conducted using Kalman filters[10]. Since wireless network and in-vehicle networks have different time delays (large vs. small), data sampling rates (slow vs. fast) and reliability characteristics (lossy vs. lossless), the traditional Kalman filter needs to be modified. For lossy networks, reference[11] provides a good survey on state estimation, closed-loop stability, and controller synthesis with sampling, delay, and packet dropouts. In order to deal with the loss of remote data, Kalman filter updating is conducted through the following strategy: when local data is reliable, the Kalman filtering is conducted solely based on local data; when remote and local data are both reliably available, the Kalman filtering is conducted by switching from local data to remote data upon the arrival of remote data, and then switching back to local data when the system is waiting for arrival of the next remote data; when both local and remote data are not reliably available, the Kalman filtering is completely turned off and a dead reckoning scheme is used.

Cloud Based Vehicle Health Monitoring

An anomaly in a car can start from a weak part, such as a *defect* in its subsystem. If the defect breaks down, the subsystem has a *fault*. Although a careful inspection can reveal the fault, the driver might not be aware of it while driving. The fault puts the subsystem into an *error* state, which may not be noticed by the driver unless there is an error indicator. The error state can lead the subsystem to *malfunction*, which further causes performance *degradation*. The degradation eventually causes the subsystem to *fail* to act properly or perform as anticipated. A system is said to be healthy if it is free from any defect, fault, error, malfunction, degradation, and failure. *Health monitoring* is a means of using electronics to detect anomalous states before failure occurs. Controlled systems (e.g., drivertrain, powertrain, and brake controls) each have their own health monitoring system. They can directly detect any or a combination of defects, faults, errors, and malfunctions using dedicated sensors or multi-purpose sensors or even performing a self-test.

There are usually no on-board electronics to directly detect an anomaly for mechanical systems; only regular service inspections. Although mechanical systems are designed to last for their life span, an anomaly can happen due to road or driving hazards. For example, a vehicle constantly driven on bad road segments can develop excessive wear in its chassis systems. Mechanical subsystems can also develop faults, errors, malfunctions, and degradation due to normal wear through normal usage. For example, the shock absorbers of a high mileage vehicle might not provide enough damping. Since a driver might not be able to sense the mechanical anomaly between regular service intervals, automatic health monitoring is desired for key mechanical subsystems. For autonomous vehicles, health monitoring will rely even more on electronics due to decreased involvement of the driver. Thus, both health monitoring and fault tolerance control are critical for the mass production of autonomous vehicles. An anomaly during normal usage can be detected either by installing new sensors (e.g., pressure sensor for tire pressure drop) or using indirect sensors (e.g, tire imbalance can be detected from the wheel speed sensor measurements used for brake control functions)[12]. The time history of mechanical system response is important since gradual degradation might be predicted from trends in historical data.

In addition to storage, many off-line diagnostic algorithms can be performed in G_r. Namely, a large chunk of data collected through the in-vehicle networks related to a certain anomaly can be uploaded to G_r and various complex diagnosis agents can be called in G_r to process the data. Certain parameters, which can characterize the anomaly pattern, might be identified and the fault models might be registered in G_r. In this way, a LSRC strategy can be used.

Cloud Based Driver State Estimator

The trend of increasing personalization, autonomy, and intelligence in automotive controls requires the creation of "driver-aware" vehicles to offer driver specific features maximizing safety, performance, and comfort, while still leaving full responsibility/control of the vehicle to the driver. This presumes well-developed algorithms for “knowing” the driver through learning the driver actions, capability, attentiveness, and driving style/behavior.

The driver might be modeled through a dynamic system with a mix of deterministic and stochastic parameters. The development of such a driver model in real-time for control is challenging. Existing approaches to this problem include: identifying deterministic driver models[13], modeling expert driver control for achieving optimal maneuvers[14], modeling driver car-following control[15], modeling driver through a model predictive control approach[16], modeling driver through Markov chains[17], real-time driver behavior identification[18], and real-time workload estimation[19].

Let's use a car-following example to illustrate how driver state can be indirectly identified. During car-following, the driver controls the vehicle to follow a leading vehicle with zero speed difference, or a constant relative distance, or the combination of the two. A simplified model can be approximated by using a proportional-derivative (PD) feedback control law[20]

$$\ddot{x}_l - \ddot{x}_f - \ddot{x}_g = -c_v(\dot{x}_l - \dot{x}_f - \dot{x}_g) - c_s(x_l - x_f - x_g) \qquad 1$$

where x_l and x_f are the travel distances of the leading and following vehicles, x_g is the desired gap, and c_s and c_v are two control gains selected by the driver. The relative velocity $\Delta v = \dot{x}_l - \dot{x}_f$ and relative distance $\Delta s = x_l - x_f$ are measured by on-board radar. The following vehicle acceleration, $\ddot{x}_f$, is measured through a longitudinal accelerometer with output a_x. Considering the time delay between the driver's actuation of brake/acceleratio ζ_p pedal and the vehicle's longitudinal deceleration/acceleration, **Equation 1** can be expressed as

$$a_{x_{k+r}} = c_v(\Delta v_k - \dot{x}_{g_k}) + c_s(\Delta s_k - x_{g_k}) + \ddot{x}_{l_k} - \ddot{x}_{g_k} \qquad 2$$

Equation 2 is a second order system, the response time t_p and the damping ratio ζ_p can be computed as soon as the control gains c_s and c_v are identified

$$t_p = \frac{2\pi c_s}{\sqrt{4c_s - c_v^2}}, \zeta_p = \frac{c_v}{2\sqrt{c_s}} \qquad 3$$

which are determined from the driver's driving behavior. Reference[20] introduces an evolving model to identify multiple sets of c_s and c_v, and uses them to compute the corresponding sets of t_p and ζ_p so as to characterize the driving behavior in real-time under different conditions, e.g. leisure, normal, and aggressive driving. The evolving modeling approach offers flexibility in developing models that match the performance of a wide variety of drivers under different circumstances, but is rather limited for on-board applications due to its evolving structure, number of clusters, and corresponding linear models that can significantly vary for different drivers. However, it fits very well into the cloud-computing paradigm since it allows storage of the models associated with different behaviors in the cloud and the creation of a detailed summary of driver behaviors over extended periods.

CLOUD BASED SUPERVISORY CONTROL AGENTS

The control computations[21, 22] can be conducted in G_r in a pseudo-real-time fashion to conform to wireless communication constraints. G_r clones its software onto multiple virtual machines at run-time for individual vehicles to access. Certain automotive controls are dictated by the models that describe the dynamics to be controlled. Errors, variations in the model parameters, and disturbances in these models negatively affect the control performance. It is known that accurate real-time parameter identification and adaptation of these models can greatly improve the control performance, for example, using the indirect adaptive control method[23]. Due to the computation and storage limitation, ECUs cannot readily perform robust real-time parameter identification for each model. Those tasks can be conducted in G_r.

The collection of multiple models, once validated through a validation process conducted in G_r, can, for example, be used for storing the historical record of the identified model parameters for health monitoring. With multiple validated models, the so-called falsified adaptive control approach might be applied[24]. Those models might also be used for designing model predictive control (MPC)[25], and the MPC agent can be run using the optimization software in G_r to find the control parameters based on the evolving models, effectively functioning as an adaptive MPC (AMPC).

Fuel Economy Optimization Agent

The fuel economy route and speed optimization problem can be summarized as follows:

Fuel Economy Route & Speed Optimization (FERSO): *Given a starting location X^0 at time t_0 and a destination location X^f with terminal time t_f, find the optimal route from all the routes connecting X^0 and X^f such that*

- *the fuel consumption of finishing the route is minimized*
- *vehicle dynamical constraints are obeyed*
- $t_f - t_0 < \Delta t$ *for a given positive real number*

and find the optimal travel speed profile such that when the vehicle uses such a travel speed profile (e.g., with cruise control), the vehicle's fuel consumption can be further optimized.

A FERSO agent solves a FERSO problem through a constrained optimization using dynamic information from the traffic, the road surface, the road geometry, the vehicle and its powertrain models, etc. Due to the complexity of constrained optimization, such a FERSO agent needs to run its computation in G_r.

In the following we consider a specific FERSO agent that optimizes the speed profile under the assumption that the route is given, and the preview of the road grade and speed limits along the route are available. The approach uses data from G_r and the in-vehicle network. In G_r, data including those from geographical information systems (GIS), traffic patterns, and speed limits are used together with vehicle models stored in G_r. All the available vehicle data such as the initial starting location, final destination, and optionally any waypoints, is transmitted from the in-vehicle networks to G_r. The optimization is conducted by using an optimization agent in G_r and the optimal velocity profile is displayed to the driver through a human-machine interface (HMI). **Figure 3** illustrates such an approach and the details can be found in[4].

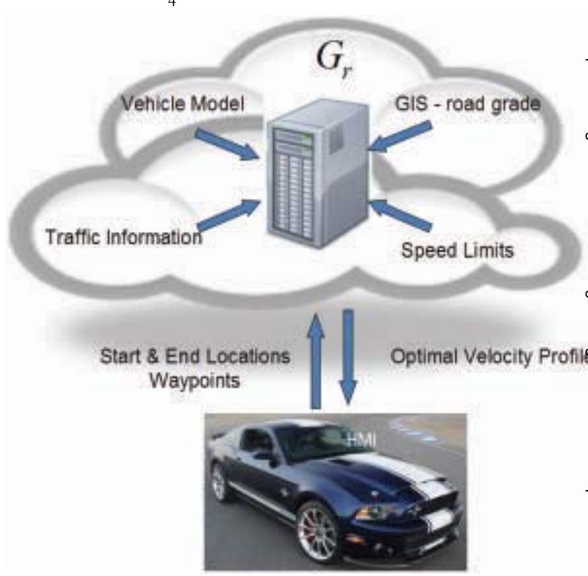


FIG. 3 Vehicle speed profile optimization using a Fuel Economy Route & Speed Optimization (FERSO) agent

Ride and Handling Optimization Agent

The traditional suspension system provides a compromise among ride control, ride comfort, and driving safety. Ride control requires limiting the sprung-mass motion within the available design constraints, ride comfort requires isolating the sprung mass from the road disturbance while meeting the available suspension space and wheel travel requirements[26], and vehicle handling and driving safety in general requires keeping the tire in contact with the road by reducing wheel hop. The suspension spring rates are designed to produce rigid body heave, pitch, and roll modes with frequencies in the range of 1 to 2 Hz. The levels of damping are designed to provide good ride comfort and at the same time to provide a certain level of ride control. For example, good secondary ride (i.e., elimination of high-frequency ripples) needs lower level suspension damping and in general an overall soft suspen-

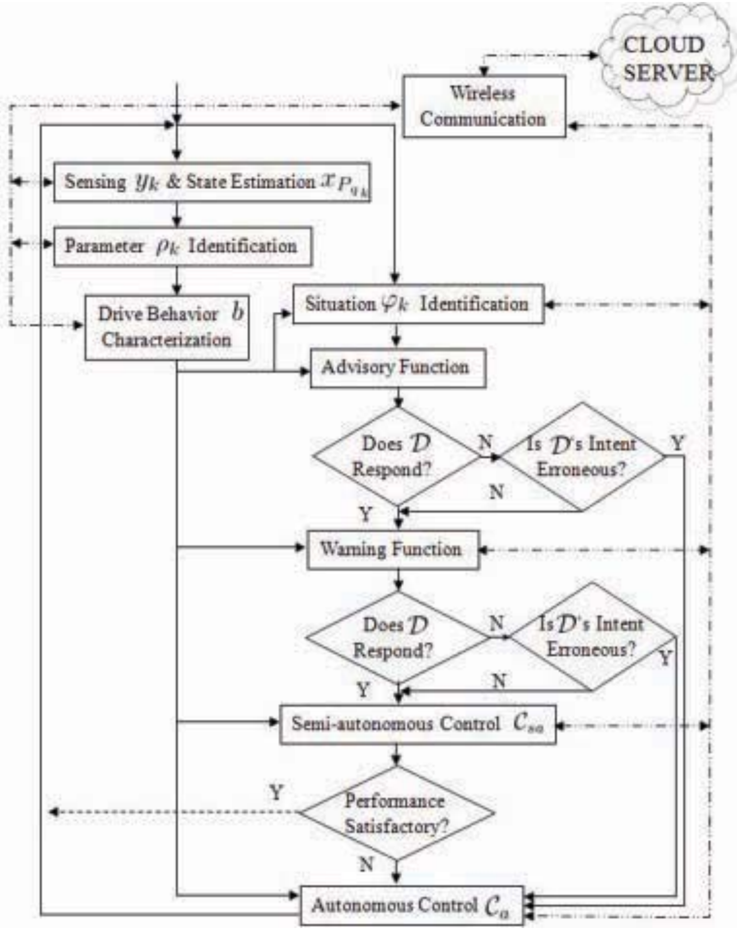


Fig. 4 **Flow-chart for DAAS functions**

sion setting. However, if it is too soft, primary ride (i.e., elimination of low-frequency bumps) and vehicle handling could suffer.

For vehicles equipped with controlled suspensions, the conflict between primary and secondary ride for traditional suspensions can be reduced through adaptation to different road and driving conditions. The suspensions are actively controlled for optimizing the vehicle's ride control, ride comfort, and vehicle handling and driving safety. Through intelligence, i.e., computers, the actively controlled suspensions can incorporate several advantageous features such as:

- **road adaptive control**
- **ride/handling feedback control**
- **road preview control**
- **abnormal road mitigation**

Majority of such suspension controllers considered so far are road adaptation and handling feedback controls[26], namely, the controller adjusts the suspension damping or stiffness or forces reactive to the measured or estimated road disturbance (e.g., road roughness) and to the driver's control of braking, throttling, or steering. Some suspension controllers take road preview data into account[27, 28], where the suspension controller is adjusted with respect to the previewed road. For abnormal road conditions such as potholes and road edges, the suspension control can take predictive actions to mitigate the consequences of driving over those abnormal road situations.

While road preview can be conducted through vision sensors, they can also use the CCS to collect crowdsourcing data from individual vehicles driving over the abnormal roads. The Ride and Handling Optimization (RHO) agent uses the preview road profiles stored in to optimize the suspension control. The control action of the RHO agent can be summarized as follows:

- if the road data in G_r shows rough terrain ahead, the RHO agent will be called to optimize the suspension control command to maximally isolate the secondary ride and at the same time to achieve a certain level of acceptable ride control

- if the road data in G_r shows rough terrain together with primary ride contents, the RHO agent will be called to optimize a balanced cost which is a function of ride comfort, the ride control, and drive safety

- if the road data in G_r shows heavy traffic in the road ahead, the RHO agent will be called to maximize driver safety (e.g., using maximum damping or suspension force to keep the wheel on road in order to prepare the vehicle for emergency maneuvering)

REFERENCES

1 Aoyama, M., 2012 "Computing for the Next Generation Automobile," *Computer*, 45(6), pp. 32-37.

2 Mell, P., and Grance, T., 2011, *NIST Definition of Cloud Computing*, Special Publication 800-145, National Institute of Standards and Technology, U.S. Dept. of Commerce.

3 Furht, B., and Escalante, A., 2011, *Handbook of Cloud Computing*, Springer.

4 Wollaeger, J., Kumar, S., Onori, S., Cairano, S. , Filev, D., Ozguner, U., and Rizzoni, G., 2012, "Cloud-Computing Based Velocity Profile Generation for Minimum Fuel Consumption: a Dynamic Programming Based Solution," *American Control Conference*.

5 Li, Z., Chen, C., and Wang, K., 2011, "Cloud Computing for Agent-Based Urban Transportation Systems," *IEEE Intelligent Transportation Systems*, 26(1), pp.73-79.

6 Ford Motor Company, 2012, *MyFord Mobile: Communicate with Your Ford Focus Electric*, http://media.ford.com/images/10031/MyFord_Mobile.pdf.

7 AUTOSAR, http://www.autosar.org/

8 Iwai, A., and Aoyama, M., 2011, "Automotive Cloud Service Systems Based on Service-Oriented Architecture and Its Evaluation," *IEEE 4th International Conference on Cloud Computing.*

9 Wang, F., 2008, "Overview of Agent-based Control and Management for NCS," *Networked Control Systems Theory and Application*, edited by F. Wang and D. Liu, Springer.

10 Grewal, M., and Andrews, A., 2001, *Kalman Filtering: Theory and Practice Using Matlab*, Wiley.

11 Hespanha, J., Naghshtabrizi, P., and Xu, Y., 2007, "A Survey of Recent Results in Networked Control Systems," *Proc. of the IEEE*, 95(1), pp. 138-162.

12 Lu, J., Filev, D., and Johnson, L., 2011, "Real-time Tire Imbalance Detection Using ABS Wheel Speed Sensors," *SAE International Journal of Materials and Manufacturing.*

13 Macadam, C., 2003 "Understanding and Modeling the Human Driver," *Vehicle System Dynamics*, 40(1-3), pp. 101-134.

14 Velenis, E., Tsiotras, P., and Lu, J., 2008, "Optimality Properties and Driver Input Parameterization for Trail-braking Cornering," *European J. of Control*, 14(4).

15 Burnham, G., Seo, J., and Bekery, G., 1974, "Identification of Human Driver Models in Car Following," *IEEE Trans. on Automatic Control*, 19(6), pp. 911-915.

16 Prokop, G., 2001, "Modeling Human Vehicle Driving by Model Predictive Online Optimization," *Vehicle System Dynamics*, 35(1), pp. 19-53.

17 Pentland, A., and Liu, A., 1999, "Modelling and Prediction of Human Driver Behavior," *Neural Computation*, 11, pp. 229-242.

18 Lu, J., Filev, D., Prakah-Asante, K., and Tseng, F., 2009, "From Vehicle Stability Control to Intelligent Personal Minder: Real-time Vehicle Handling Limit Warning and Driver Style Characterization," *Proc. of IEEE Symp. Series on Comp. Intelligence.*

19 Prakah-Asante, K., Filev, D., and Lu, J., 2010, "Hybrid Intelligent System for Driver Workload Estimation for Tailored Vehicle-Driver

- if the road ahead has a pothole, the RHO agent will be called to conduct abnormal road mitigation, e.g., adjust the damping and stiffness of suspensions or reposition the vehicle attitudes in order to minimize the effect of driving over the pothole

The RHO agent might also be used with other automotive controls to enhance other aspects of vehicle performance.

Driver Assistance and Active Safety Agent

Driver-assistance (DA) systems monitor, classify, and determine driving conditions where a driver needs to be reminded of potential danger or be assisted with additional control authority to enhance his control or reduce his workload. The driving conditions are classified based on driver intent, driving scenario, deviation from normal driving, and driver control action. A suite of DA features can currently be found in vehicles[29] including adaptive cruise control, lane departure warning, lane keeping aid, driver alert, auto high beam, traffic sign recognition, active park assist, blind spot information system, hill start assist and speed limiter. Other DA features, such as lane change assistance, intelligent speed advice, night vision, driver drowsiness detection, hill descent control, traffic jam assist[30], are also being developed.

Active safety (AS) features involve autonomous actions, which are one step further from DA. Such AS features focus on avoiding accidents or mitigating injury/damage due to unavoidable accidents through various vehicle controls. Notable AS features in production[31] include electronic stability control (ESC)[32], roll stability control[33], collision mitigation by braking, curve control, emergency steering assist[34-35], and multi-collision braking[36-37].

The combination of DA and AS is denoted as driver assistance and active safety (DAAS). Due to the involvement of drivers and driving scenarios, DAAS features need to be both scenario-aware and driver-aware. Our focus here is on the *DAAS agents* that use data from G_r and from in-vehicle networks to help identify conditions to support or enhance the traditional DAAS features.

Figure 4 shows the action flow of the operations used in a DAAS agent, which combines data y_k from on-board sensors, estimated plant state x_{p_k}, and vehicle, driver, and community states. The parameter ρ is first deduced from on-board data, state estimation, and data from G_r. The driver behavior is next determined from sensor data, previous calculations, and cloud data representing the recorded driver behavior. The driving condition φ is also determined from on-board and cloud data. If a condition in which the driver needs to be reminded is present, the DAAS agent first sends out advisory information presented through an HMI device. If a condition where driver attention is not compatible with the demand on the driver (after the advisory display), the DAAS agent will issue a warning signal. Based on the way in which the driver responds to the advisory or warning info, the DAAS agent determines further actions. For example, if the driver is not responding to the warning signal, the agent will initiate a semi-autonomous action through C_{sa} that can take aggressive control measures to operate the vehicle but still tries to follow the driver's general intent. If after the C_{sa}'s action, the motion of the vehicle does not achieve the desired safety level, the agent will initiate an autonomous controller C_a that can override the driver's intent and automatically control the vehicle (e.g., either stopping the vehicle or reducing the vehicle speed). While the decision making itself can be performed on ECUs, much of the information used for the decision making such as the state of the driver, the vehicle, the road, the traffic, etc. and their utilizations need to be performed in G_r.

The proposed DAAS agents can be used for many scenarios. For example, a vehicle path anomaly might be determined by comparing the host vehicle trajectory with the community trajectory deduced from all surrounding vehicles; the safe path to which the vehicle can escape when there is an emergency or an avoidable accident or after the vehicle has been engaged in a collision can be deduced from cloud information about the traffic together with the local sensor measurements. Another example is that all driver assist features try to warn the driver without knowing if the driver prefers this or not and DAAS features can be personalized by utilizing the learned driver behavior registered in G_r. A driver behavior centric approach is required if a warning needs to adapt to driver at any specific time during driving.

Crowdsourcing Agents

Individual vehicles can sense road and traffic conditions using on-board sensors. The sensed data can then be uploaded to G_r and stored under specific labels. The average behavior of all the vehicles traveling within close time and spatial proximity is called

Communication and Interaction," *Proc. of IEEE Conf. on Systems, Man, & Cybernetics.*

20 Filev, D., Lu, J., Tseng, F., and Prakah-Asante, K., 2011, "Real-time Driver Characterization During Car Following Using Stochastic Evolving Models," *Proc. of IEEE Conf. on Systems, Man, & Cybernetics.*

21 Hrovat, D., Tseng, H., Lu, J., Deur, J., Assadian, F., Borrelli, F., and Falcone, P., 2011 "Vehicle Controls," *The Control Handbook: Control System Applications*, William S. Levine.

22 Hrovat, D., Jankovic, M., Kolmanovsky, I., Mahner, S., and Yanakiev, D., 2011, *Powertrain Control*, The Control Handbook: Control System Applications, William S. Levine.

23 Ioannou, P., and Fidan, B., 2006, *Adaptive Control Tutorial,* SIAM.

24 Safonov, M., and Tsao, T., 1997, "The Unfalsified Control Concept and Learning," *IEEE Trans. on Automatic Control*, 42(6), pp. 843-847.

25 Hrovat, D., Cairano, S., Tseng, H., and Kolmanovsky, I., 2012, "The Development of Model Predictive Control in Automotive Industry: a Survey," *IEEE Multi-Conference on Systems and Control.*

26 Hrovat, D., 1997, "Survey of Advanced Suspension Developments and Related Optimal Control Applications," *Automatica,* 33(10), pp.1781-1817.

27 Tomizuka, M., 1976, "Optimum Linear Preview Control with Application to Vehicle Suspension - Revisited," *Trans. of ASME on J. of Dynamic Systems, Measurement and Control,* 98(3), pp.309-315.

28 Hrovat, D., 1991, "Optimal Suspension Performance for 2-D Vehicle Models," *Journal of Sound and Vibration,* 146(1), pp. 93-110.

29 Ford Motor Company, 2011, *Ford Focus and C-Max Customers Take New Technologies Onboard,* http://media.ford.com/news/fordfocusandc-maxcustomerstakenewtechnologiesonboard.htm.

30 Ford Motor Company, 2012, *Ford Develops Traffic Jam Assist and New Parking Technology to Help Address Future Mobility Challenges,* http://media.ford.com/article_pdf.cfm?article_id=36722.

31 Ford Motor Company, *Ford, Volvo and Jaguar Developing 'Intelligent' Safety Systems of Tomorrow,* http://media.ford.com/article_display.cfm?article_id=22078.

32 Tseng, E. Ashrafi, B., Madau, D., Brown, T., and Recker, D., 1999, "Development of Vehicle Stability Control at Ford", *IEEE/ASME Transactions on Mechatronics,* 4(3), p223-234.

33 Lu, J., Messih, D., Salib, A., and Harmison, D., 2007, "An Enhancement to an Electronic Stability Control System to Include a Rollover Control Function," *SAE Transactions,* 116, pp. 303-313.

34 Floyd, F., 2010, "Continental Wants Emergency Steer Assist to Drive Cars Away from Accidents," *Motor Trend*, Jun. 16.

35 Chakraborty, I., Tsiotras, P., and Lu, J., 2011, "Vehicle Posture Control through Aggressive Maneuvering for Mitigation of T-Bone Collisions," *IEEE Conference on Decision and Control*.

36 Bickerstaffe, S., 2012, "Conti Puts the Brakes on Secondary Collisions: Emergency Braking Function Stops the Car after the First Impact," *Automotive Engineer*, July 19.

37 Zhou, J., Lu, J., and Peng, H., 2010, "Vehicle Stabilisation in Response to Exogenous Impulsive Disturbances to the Vehicle Body," *International Journal of Vehicle Autonomous Systems*, 8(2/3/4), pp. 242-262.

the *community state*. A cloud supervisory agent, called a *crowd-sourcing agent*, can be used to classify and summarize the data from individual vehicles to assemble community states for a specific location. For example, it computes the minimum, maximum, and average speeds traveled by individual vehicles and the speed frequency distribution at a specific location.

The road condition can be similarly evaluated. Presently, detailed road maps with updated road characteristics are not available. Although a road surface and geometry map can be obtained by using a laser scanner, its practical usage is rather limited. Thus, multiple vehicles traveling and collecting data on the same roads provides an opportunity for "digitally mapping" the roads. The real-time measurements from in-vehicle networks can capture road features including roughness, geometry variations, friction levels, and curvature variations. Notice that if the road sensing vehicles only use their own sensed data, its usefulness is very limited since there is no preview type of information unless the vehicle drives on a previously classified road. If all the vehicles passing through a road segment are sending their data to G_r, the road conditions can be much more accurately determined. A particular vehicle's data can be used for preview by other vehicles. The road surface friction level might be determined through variables calculated for anti-lock braking, traction control and ESC from individual vehicles. The road geometry such as road bank and slope can be determined from the on-board inertia sensors and sent to G_r.

VISION FOR FUTURE MOBILITY

It is expected that increasing numbers of people will live in big cities, or megacities, with highly concentrated populations. Drivers navigating their way in such increasingly crowded urban areas will need vehicle control systems with an increased level of intelligence such as fast data processing, adaptation, control reconfiguration, optimization in reaction to massive incoming information. Vehicles will also be used as personalized information channels to find the best services and sources for the driver. Such needs will lead to further advancement of automotive controls. This paper considers one possible transformational direction: integrating vehicle controls with cloud computing for enhancing their informational and computational capability. Future vehicle controls can be expected to be personalized and adaptive to driver needs, capability, and preferences (e.g., aged drivers need more driver assistance features which might be annoying to young drivers, experienced driver might prefer a different control calibration). This personalization is transferrable to autonomous driving by designing the autopilot to mimic human drivers. Adaptive controls are likely to be one of the most useful strategies for such personalization.

In future vehicle controls, the cloud can be used for very demanding computations that otherwise cannot be accomplished by on-board ECUs, especially for information-intensive tasks. The so-called local-simple-remote-complex vehicle control strategies are likely to unlock the potential of implementing methods and tools, e.g. dynamic programming, that are presently used only in an off-line setting. The cloud can also be used as a storage place to record current and historic vehicle data that can be used for predictive diagnosis and prognostics of the vehicle health. ■

ABOUT THE AUTHORS

JIANBO LU received his Ph.D. degree in Aerospace Engineering from Purdue University in 1997. He joined Ford in 2000, where he is currently a Technical Expert. His interests are in automotive controls, intelligent & adaptive vehicles, driver assistance, and active safety. He is a recipient of the Henry Ford Technology Award. He is a co-inventor on more than 70 US patents and is the author/co-author of more than 60 technical papers. He is currently serving as an Associate Editor for IFAC J. of Control Engineering Practice and IEEE Trans. on Control Systems Technology and is the Chair of Intelligent Vehicular Systems & Control Technical Committee, IEEE SMC Society. Dr. Lu is a senior member of IEEE and a member of ASME, SAE, and Tau Beta Pi. The photograph below shows Dr. Lu at the recent FISITA 2012 conference in Beijing.

DAVOR HROVAT received his Ph.D. degree in Mechanical Engineering from the University of California, Davis in 1979. Since 1981, he has been with the Ford Motor Company where he conducted and led R&D efforts on various aspects of chassis, power train and overall vehicle control. He is a co-inventor on more than 50 US patents and is the author/co-author of more than 100 technical papers. In 2006 he was appointed the first Henry Ford Technical Fellow in the area of Controls. He is a Fellow of the American Society of Mechanical Engineers and the recipient of the 1996 ASME/Dynamic Systems and Control Innovative Practice Award and the 1999 AACC Control Engineering Practice Award. In addition, Dr. Hrovat is a Fellow of IEEE, IFAC, and a member of the National Academy of Engineering.

DIMITAR FILEV is Senior Technical Leader, Ford Research & Advanced Engineering. He received his PhD. degree in Electrical Engineering from the Czech Technical University in Prague in 1979. He joined Ford Motor Company in 1994. His research interests are in modeling and control of complex systems, intelligent control, fuzzy and neural systems, and their applications to automotive engineering. He has published 4 books, over 200 technical papers, and holds 20 US patents. He is the recipient of the 2008 Norbert Wiener Award of the IEEE SMC Society and the 2007 IFSA Outstanding Industrial Applications Award and is Co-Editor in Chief of the Int. J. on Evolving Systems. Dr. Filev is a Fellow of IEEE and IFSA and is shown in the photograph below delivering a lecture at the 2009 Benjamin Franklin Symposium at Villanova University.