Modeling a Physical System for System Control and Calibration

Introduction

Processes that involve physical systems are difficult to optimize if the underlying system is too complex to be understood or simulated accurately. In that case, the calibration of the physical system is expensive due to hand-tuning of the system's operating parameters, i.e. actuators, by domain experts. One example is robust control of a robot which is hard to tune manually due to the high number of actuators involved. Another example from the automotive domain is optimization of a combustion engine where a huge number of parameters with non-linear cause-effect relationships play a role in the system's response. In practice, today's software functions to control a combustion engine are still based on look-up tables that are at most three dimensional, severely limiting their applicability in high-dimensional scenarios. In an engine control unit, resulting inaccuracies in the function output can lead to additional fuel consumption, drivability deficiencies, and in the worst case to engine damage.

In order to avoid manual tuning of system parameters, we use machine learning methods to approximate the physical system. If the mathematical model can approximate the physical system accurately, model-based control mechanisms can be derived to optimize the process at execution time.

Toolchain from Measurement Planning to System Control and Calibration

To enable automatic calibration or control of a physical system, such as a robot or a vehicle engine, our goal is to create a model that learns all characteristics of the underlying system from measurements only. This toolchain is illustrated in Figure 1. To learn an accurate data-based model, relevant system inputs are identified first. Second, a measurement plan is compiled, describing for which combination of inputs the system responses have to be measured. After collecting these measurements from the real system, a machine learning model is trained to imitate the system behavior based on these measurements. Such a model can finally be applied either for

- **system calibration:** the parameters of the physical system are tuned using the machine learning model as an approximation to the system's output. One practical example from the automotive domain is to determine the optimal ignition timing setpoint to maximize the efficiency of a combustion engine.
- **system control:** the predictions of the machine learning model are treated as a virtual sensor of the system and decisions are based on these predictions. Physical sensors are costly, sometimes unreliable, and need to be maintained. In addition, their operating range is in general restricted. In that case, virtual sensors may replace physical sensors, convert discrete sensor measurements to continuous signals, or provide a redundant signal to a physical sensor, thus enabling the diagnosis of faults and serving as a substitute or fallback solution.

  To return to the combustion engine example, the physical temperature sensor in the exhaust gas tract can, due to cost reasons, be replaced by a machine learning model that predicts the exhaust gas temperature. This prediction in turn can be utilized for engine control to protect the components of the exhaust gas tract.

  In practice, for efficient system control, it may be necessary to create specialized hardware to guarantee fast model evaluations.

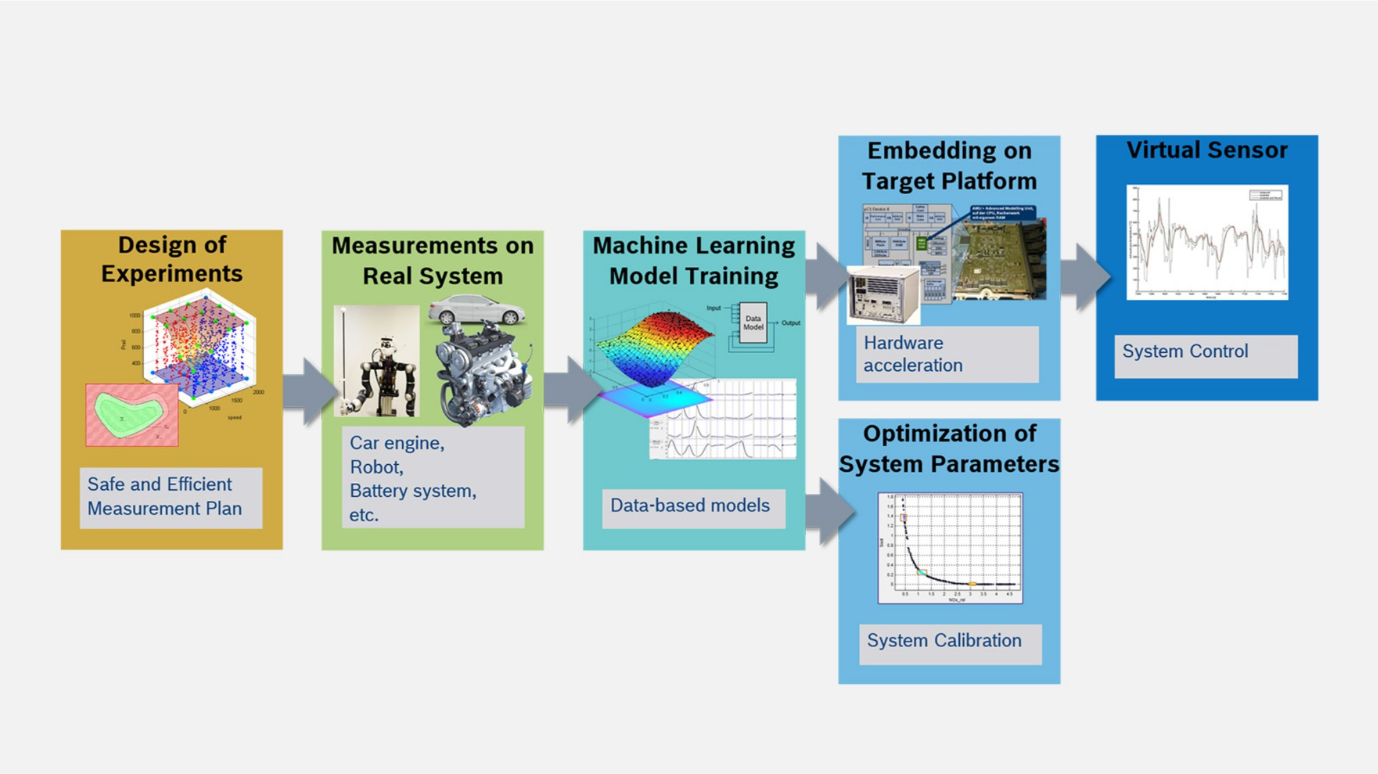

Figure 1: Simplified toolchain to model a physical system for system control and calibration

Our Research Results

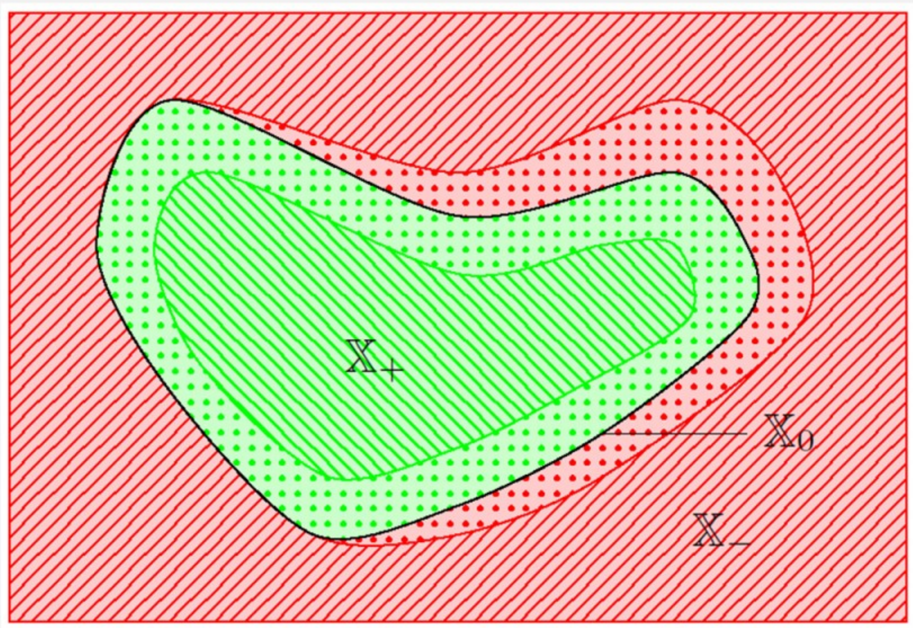

Figure 2: The input space X is partitioned into a safe explorable area X+ and an unsafe region X-. A discriminative function is learned to find the unknown decision boundary X0 between these regions. [Image source: ECML2015]

Our research in this context mainly focuses on the following four elements:

1. Design of Experiments and System Measurements: Typically, a domain expert first identifies the system inputs that influence the target system response. Second, we want to determine for which combinations of these inputs the response of the system needs to be measured to get an accurate machine learning model and, at the same time, we want to keep measurement efforts at a minimum.

For example, automotive experts know that the ignition timing setpoint together with engine speed and load influence the efficiency of a combustion engine, which in this case is the target system response. To determine the optimal ignition timing setpoint for a given engine speed and load, these inputs are varied and the level of efficiency is measured to build a data basis for the remaining toolchain.

In practice, however, not all combinations of inputs are feasible and in fact, some of them may even destruct the physical system. To address this challenge, we feedback system measurements into the sampling scheme to learn boundaries online in which it is safe to execute the planned measurements, see Figure 2. We leverage this algorithm at engine test benches for increased process stability and reduced planning effort.

Moreover, we developed methods that successfully integrate updated knowledge about the machine learning model into the Design of Experiments step. In this way, the measurement plan is continuously adapted, resulting in Safe Exploration for Active Learning algorithms [ECML2015].

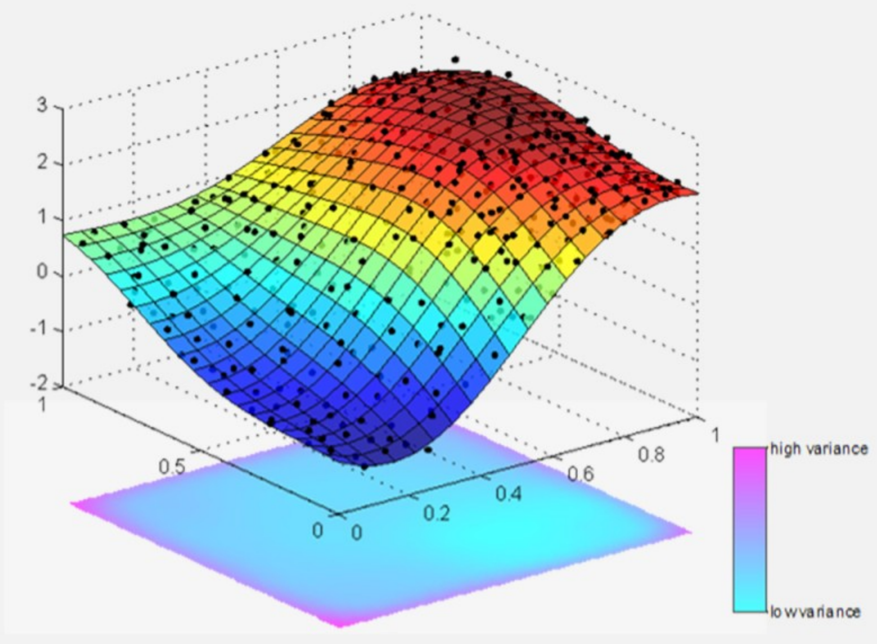

Figure 3: Gaussian process regression with measured data (black dots), model prediction (surface grid) and estimated model uncertainty (gradient from magenta to blue).

2. Machine Learning Model: After taking measurements according to the sampling scheme above, the system is modeled based on this data: we cast the physical system as a continuous vector-valued function f(x) of system inputs x, the output of which is corrupted by random measurement noise ε. In other words, the system responses y at input x are assumed to follow the form: y=f(x)+ε. The machine learning task, to estimate this unknown function f(·) from looking at N measurements of input-output pairs (x,y), is known as regression. When such a function f(·) is found, the machine learning model can make predictions for the system's outputs for any arbitrary input x.

The literature of regression methods ranges from linear regression over deep learning based methods to Gaussian process regression. For the tasks we consider in our practical applications, Gaussian process regression has proven to be favorable due to their natural smoothness assumptions and the estimation of the model uncertainty (see Figure 3). The latter can be utilized to validate whether the prediction of the model should influences the decision process or the prediction should be discarded due to high uncertainty. Moreover, these regression algorithms do not only consider inputs at the current time point, but also take into account the temporal history of inputs.

To apply Gaussian process regression in practice the models need to be extended to cope with large datasets during model training through sparsification of the model. Research advancing sparse Gaussian process regression has made significant progress in recent years. We were able to contribute relevant developments for practical use through an efficient active set selection algorithm [ICRA2015, Neuro2016].

Currently, our research focuses on making modeling of dynamical systems more efficient.

3. Embedding on Target Platform: Integrating the learned models onto a physical system is often necessary for system control. However, time restrictions such as real-time requirements can often only be met if specialized hardware accelerates model evaluation.

For the new generation of engine control units (Bosch MDG1, from device 3 upwards), we developed a module (see Figure 4) which allows to evaluate Gaussian process regression models in real time. This module, the so-called Advanced Modeling Unit (AMU), an arithmetic logical unit with separate 32kB RAM, enables fast summations of weighted exponential functions resulting in a reduction of computation time by 98% with IEEE754 accuracy (32-bit). With the advent of dedicated machine learning hardware on control units, the limitations imposed by the conventional three-dimensional look-up tables can be overcome, and Gaussian process regression can be used to directly model high-dimensional data accurately.

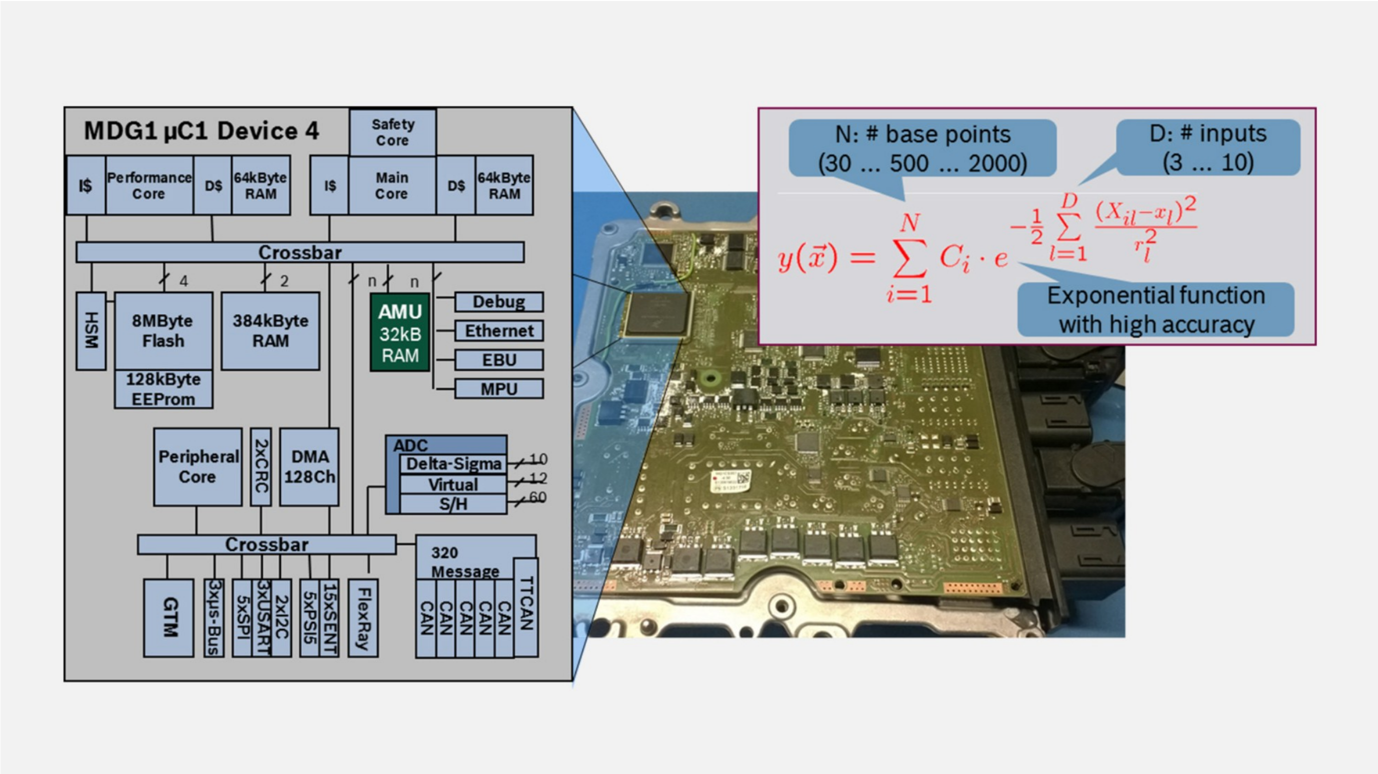

Figure 4: The Advanced Modeling Unit (AMU) on Bosch's engine control unit.

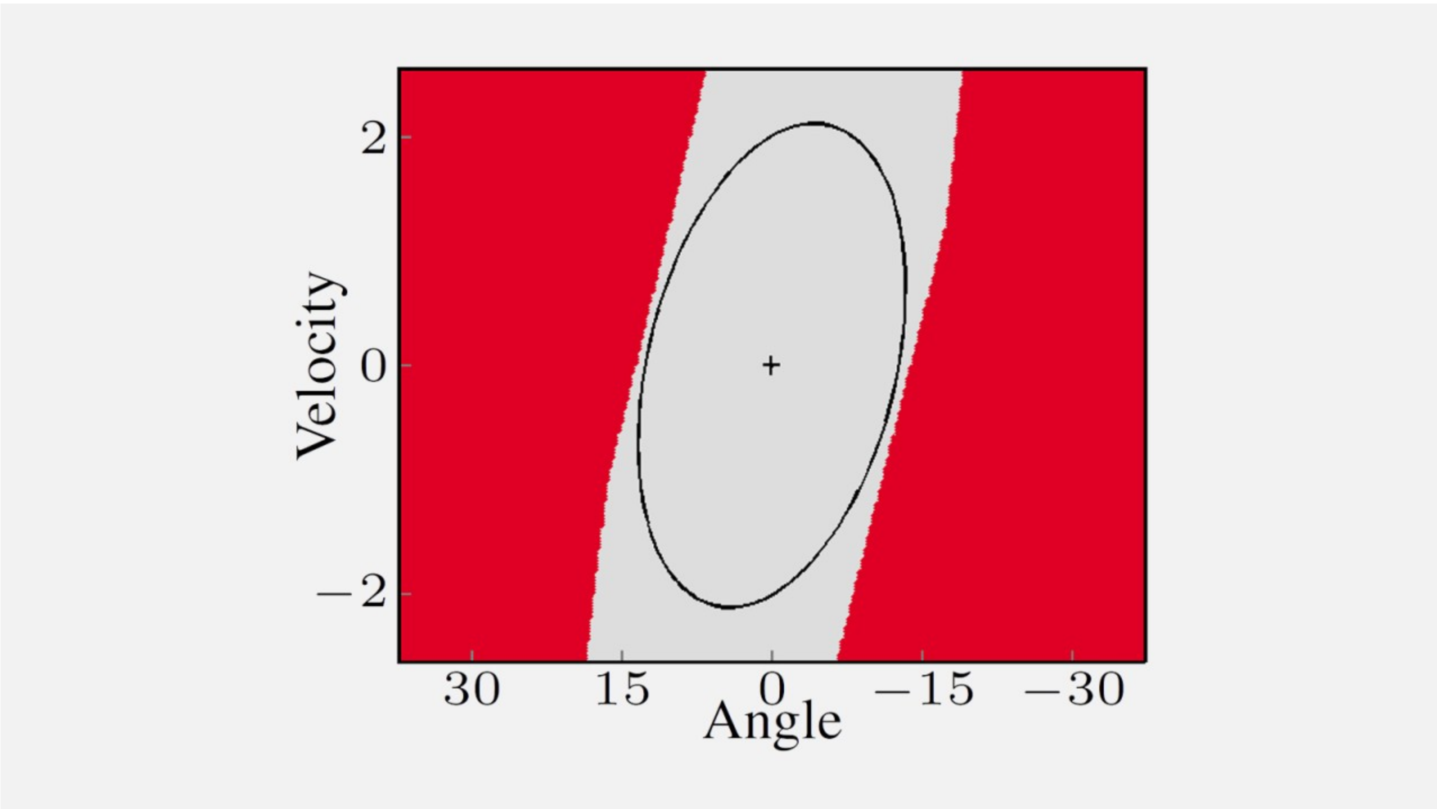

Figure 5: For a given problem, here: inverted pendulum, with the two input parameters angle and velocity, grey and red regions show points which are true stable and instable regions. All points within the ellipsoid returned by our algorithm are guaranteed to converge [source: ICML2016]

4. Model-based System Control: After learning and implementing the machine learning model on the real system, the model can be used to support control of the system. We examined the reliability of controllers which base their decisions on Gaussian process regression models to derive stability guarantees [ICML2016], see Figure 5. Furthermore, we developed and successfully learned models for robot control from little training data [ICRA2014], from large datasets with time restrictions [ICRA2015], algorithms for model-based policy learning for system control under given constraints [ECML2014], and tuning of multi-variate controllers without prior knowledge on system dynamics as developed within the CyberValley collaboration [ICRA2017].

Our Research Applications

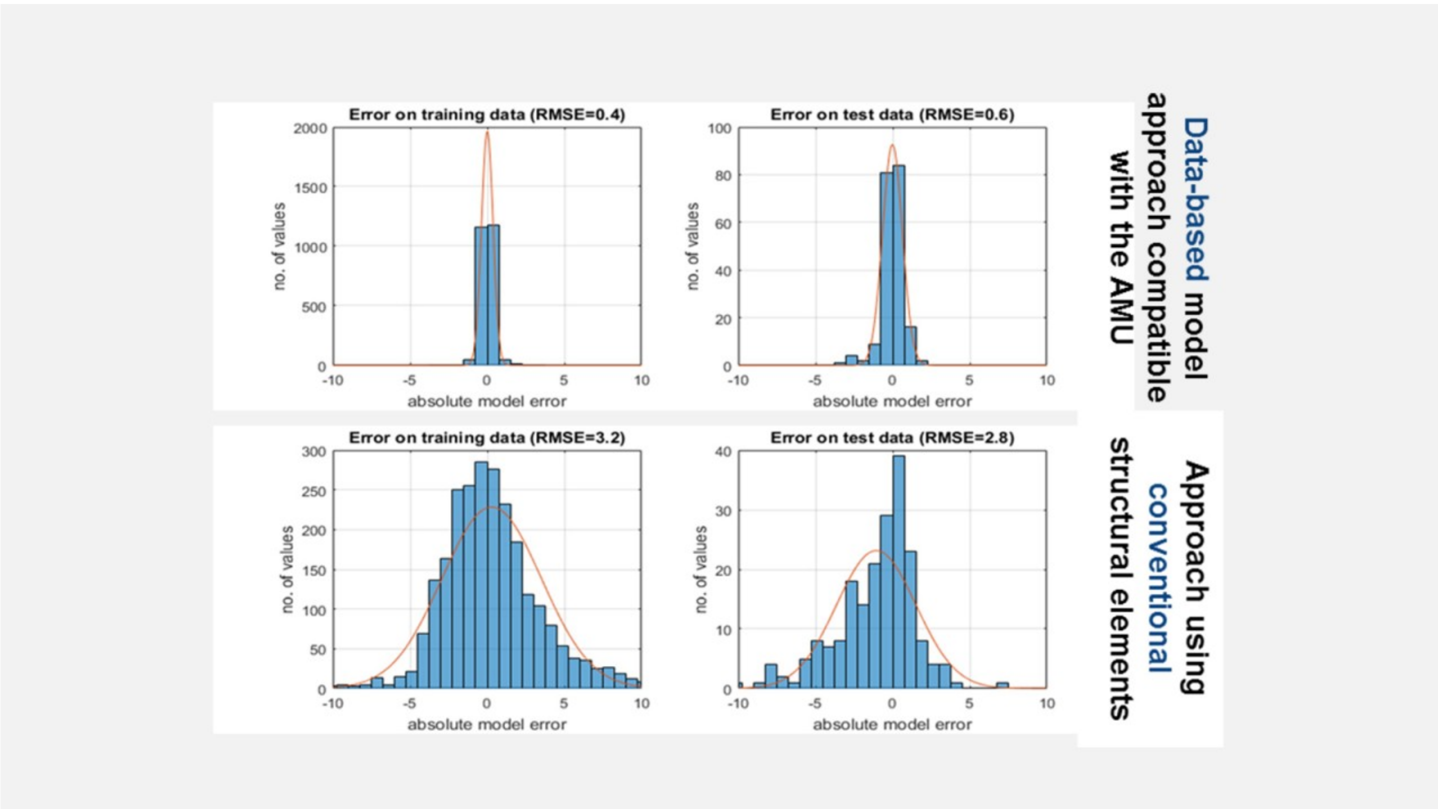

Figure 6: Performance comparison between conventional function structures on engine control units compared to the approach based on Gaussian process regression. [Image source: IAV2017]

The entire toolchain explained above has been deployed as part of the commercial product ETAS ASCMO. This tool has become state-of-the-art in the automotive domain and is used by major car manufacturers.

Furthermore, the AMU, which computes Gaussian process regression models in real-time, is an integral part of Bosch MDG1 passenger car engine control units, allowing the use of these models as virtual sensors or as pre-control structures.

We applied this toolchain for various embedded control unit functions ranging from virtual sensors for exhaust temperature, torque, or exhaust emission to pre-control structures such as the ignition timing setpoint [IAV2017]. The significant increase of ignition timing accuracy (from 2.8 degrees to 0.8 degrees crank angle standard deviation) in complex engines, is expected to reduce fuel consumption by 1-2% under real driving conditions, see Figure 6.

References

[ECML2014] Bischoff, B., Nguyen-Tuong, D., Koller, T., Markert, H., & Knoll, A. (2014). Learning throttle valve control using policy search. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML 2014), (pp. 49-64).

[ICRA2014] Bischoff, B., Nguyen-Tuong, D., van Hoof, H., McHutchon, A., Rasmussen, C. E., Knoll, A., Peters, J. & Deisenroth, M. P. (2014). Policy search for learning robot control using sparse data. In IEEE International Conference on Robotics and Automation (ICRA 2014), (pp. 3882-3887).

[ECML2015] Schreiter, J., Nguyen-Tuong, D., Eberts, M., Bischoff, B., Markert, H., & Toussaint, M. (2015). Safe exploration for active learning with Gaussian processes. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML 2015) (pp. 133-149).

[ICRA2015] Schreiter, J., Englert, P., Nguyen-Tuong, D., & Toussaint, M. (2015). Sparse Gaussian process regression for compliant, real-time robot control. In IEEE International Conference on Robotics and Automation (ICRA 2015), (pp. 2586-2591).

[ICML2016] Vinogradska, J., Bischoff, B., Nguyen-Tuong, D., Romer, A., Schmidt, H., & Peters, J. (2016). Stability of controllers for Gaussian process forward models. In International Conference on Machine Learning (ICML 2016), (pp. 545-554).

[Neuro2016] Schreiter, J., Nguyen-Tuong, D., & Toussaint, M. (2016). Efficient sparsification for Gaussian process regression. Neurocomputing, 192, 29-37.

[IAV2017] Angermaier, S., & Gopalakrishnan, M. (2017). Implementation of data-based models using dedicated machine learning hardware (AMU) and its impact on function development and the calibration processes. In International Calibration Conference – Automotive Data Analytics, Methods, DoE (IAV 2017).

[ICRA2017] Doerr, A., Nguyen-Tuong, D., Marco, A., Schaal, S., & Trimpe, S. (2017). Model-Based Policy Search for Automatic Tuning of Multivariate PID Controllers. In IEEE International Conference on Robotics and Automation (ICRA 2017).

Academic Collaborations

Learn more ›

Publications

Learn more ›