Calibration of AD/ADAS simulator: ABC method using a surrogate model

Clara CARLIER*[1][2], Arnaud FRANJU[2], Matthieu LERASLE[1]
clara.carlier@ensae.fr, arnaud.franju@renault.com, matthieu.lerasle@ensae.fr
[1] CREST - IP PARIS - 91120 Palaiseau, France [2] Renault Group - DEA-TDV - 78280 Guyancourt, France

Abstract

Recent developments in Autonomous Driving (AD) and Advanced Driver-Assistance Systems (ADAS) require an increasing number of tests to validate these new technologies. Conducting these tests on track would be too time-consuming, so automotive groups rely on simulators to perform most of the testing.
To integrate simulations into the certification process, a digital twin of the physical autonomous vehicle is created and must be calibrated to generate data that is sufficiently similar to the on-track tests.

In this work, we present an efficient methodology that will assess the quality of the simulator by comparing it to real on-track data, then calibrating and readjusting it. Once calibrated, the simulator can generate a more realistic time series. The process amounts to solving an inverse problem with an ABC method by integrating the use of a surrogate model that replaces the simulator, which is much faster and less expensive to run on specific tasks.

Introduction

- **Context:** validation and certification of Autonomous Driver (AD) and Advanced Driver-Assistance Systems (ADAS)
  - numerous onboard sensors in cars
    ▷ a large amount of information
  - many strict regulations
    ▷ a lot of on-track tests over long distances
- **Proposed solution:** develop digital platforms to model AD/ADAS and create simulations
  ▷ complete or even replace the real on-track tests
- **Problematic:** ARE THE SIMULATIONS SUFFICIENTLY CORRELATED WITH THE REAL TESTS TO BE USED LEGALLY?
- **Goal:** simulator calibration
  - integrates the simulations into the certification process by generating data similar enough to the on-track tests
  - develop a methodology that will gauge the quality of the simulator to calibrate and readjust it
    ▷ combination of the resolution of an *inverse problem* and a *direct problem*
- **Data and tools available:**
  - simulator platform: access to the SCANeR simulation software to create the desired data
  - simulated data: as numerous as wanted
  - real on-track test data: a small number

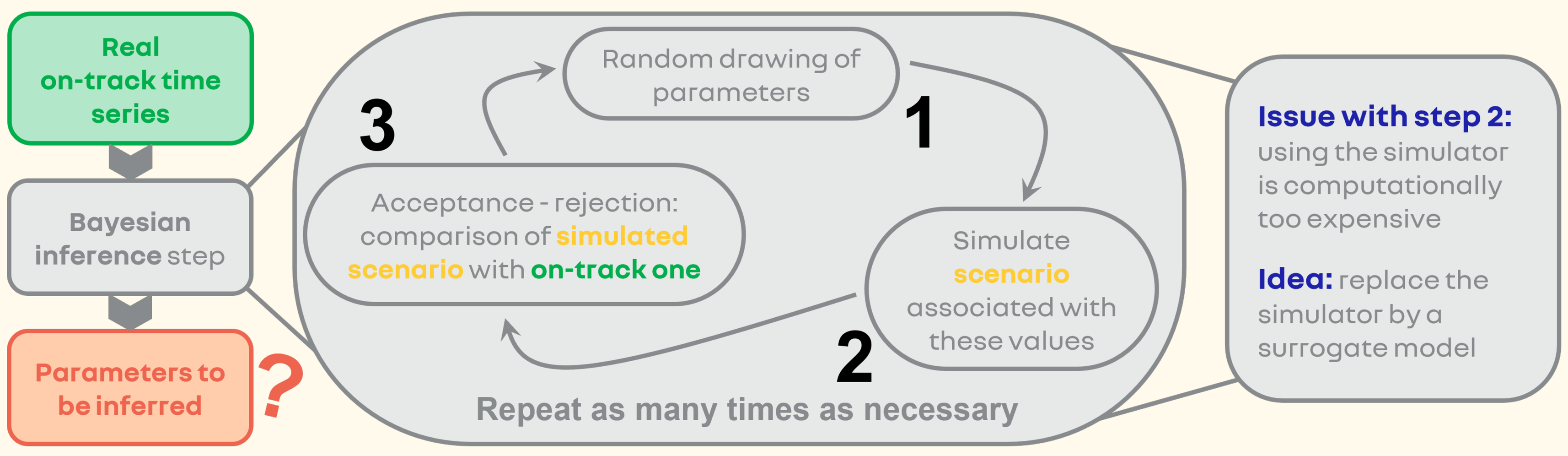

Figure 1: General process summary. The three blocks on the left represent the inverse problem which consists in finding the values of the input parameters associated with the reference on-track test. The middle section describes how ABC methods work. The last part concerns the issue and how we intend to solve it.

Functioning of the simulator S

$$S(\theta) = y$$

- **Inputs** θ**:** require different input parameters to define the desired experiment
  ▷ initial speed, braking efficiency, ..., etc.
- **Outputs** y**:** generate the associated time series describing vehicles' behavior
  ▷ speed, acceleration, ..., etc.

Inverse problem

- **We have:**
  - one so-called reference test, on-track time series y_φ
  - its associated input parameters called nominal values θ_0
- **We want:** to recover the input parameters that would simulate the *closest* time series to the reference ones
- **How to do it?**
  - Approximate Bayesian Computation (ABC) ▷ likelihood-free inference schemes
- **Problem:** each step is repeated iteratively and step 2 requires the use of the simulator which is computationally too expensive
  ▷ DEVELOPMENT OF A SURROGATE MODEL THAT MIMICS AND REPLACES THE SIMULATOR

Surrogate model $\widehat{S}$

To build the surrogate model, we construct a predictor $\widehat{S}$ which generates output time series for a given set of parameters $\widehat{S}(\theta) = y$

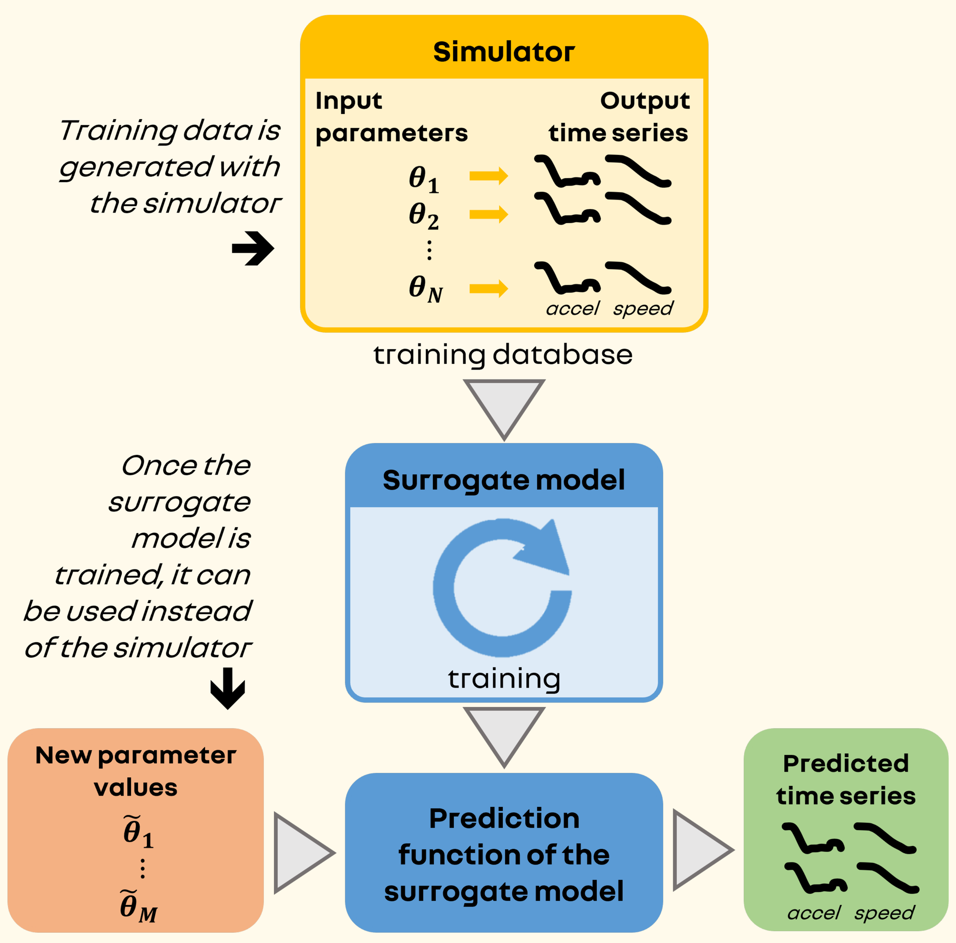

Figure 2: Summary of the training and predicting process of the surrogate model

Solving the inverse problem

- **We have:** a predictor $\widehat{S}$, a reference test y_φ and its nominal values θ_0
- **We want:** to recover the posterior distribution, by Bayes' formula

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

where the likelihood $p(y|\theta)$ is computed with $\widehat{S}$ and $p(\theta)$ is the prior distribution depending on θ_0 noted π_0

Algorithm 1 ABC acceptance/rejection method

```
Input: initial tolerance ε, distance d, prior distribution π_0
Output: Θ which contains several vectors of accepted parameters
  while nb_accepted > 0 do
    nb_accepted = 0
    for i ∈ {1, ..., 500} do
      random drawing of candidate parameters θ' ~ π_0
      generation of associated time series y' = Ŝ(θ')
      if d(y', y_φ) < ε then
        nb_accepted = nb_accepted + 1
        θ' is accepted and saved as a new value in Θ
      end if
    end for
    calculation of θ_mean, the average of all accepted θ contained in Θ
    generation of associated time series y_mean = Ŝ(θ_mean)
    update of the tolerance ε = min {ε, d(y_mean, y_φ)}
  end while
```

Obtained results

- **Obtained solutions:**
  - computation of $\widehat{\theta}$ using the Θ set output of Algorithm 1
  - selection of θ_{sim} in the training database
    ▷ the θ_i which generates the best simulation $S(\theta_i)$
- **Quality of the results:** comparison of $S(\theta_0)$, $S(\theta_{\text{sim}})$, $\widehat{S}(\widehat{\theta})$ and $S(\widehat{\theta})$

		RMSE
simulation with nominal values	$S(\theta_0)$	0.350
the best simulation in training dataset	$S(\theta_{\text{sim}})$	0.310
prediction with the inverse problem result	$\widehat{S}(\widehat{\theta})$	0.263
simulation with the inverse problem result	$S(\widehat{\theta})$	0.348

Table 1: RMSE results

- **Good results:** the ABC algorithm allows to beat the score of the nominal values and even to beat the best simulation with the surrogate model
- **Limitations and future improvements:** the parameters output by the algorithm does not allow the simulation of clearly better time series