A real-time state estimation for an electric race car

Research Project / T3100

for the study program
Computer Science

at the
Baden-Wuerttemberg Cooperative State University Stuttgart

by
Dominik Stiller

Submission Date	June 8, 2020
Thesis Supervisor	Marco Busch
University Supervisor	Prof. Dr. Zoltán Ádam Zomotor
Matriculation Number, Course	4369179, TINF17A

Declaration of Authorship

I hereby declare that the thesis submitted with the title *A real-time state estimation for an electric race car* is my own unaided work. All direct or indirect sources used are acknowledged as references.

Neither this nor a similar work has been presented to an examination committee or published.

Sindelfingen	June 8, 2020	
Place	Date	Dominik Stiller

Confidentiality Clause

This thesis contains confidential data of *DHBW Engineering Stuttgart e.V.* This work may only be made available to the university supervisor. Any publication and duplication of this thesis–even in part–is prohibited.

An inspection of this work by third parties requires the expressed permission of the author and *DHBW Engineering Stuttgart e.V.*

Abstract

Real-time computer vision applications with deep learning-based inference require hardware-specific optimization to meet stringent performance requirements. Frameworks have been developed to generate the optimal low-level implementation for a certain target device based on a high-level input model using machine learning in a process called autotuning. However, current implementations suffer from inherent resource utilization inefficiency and bad scalability which prohibits large-scale use.

In this paper, we develop a load-aware scheduler which enables large-scale autotuning. The scheduler controls multiple, parallel autotuning jobs on shared resources such as CPUs and GPUs by interleaving computations, which minimizes resource idle time and job interference. The scheduler is a key component in our proposed Autotuning as a Service reference architecture to democratize autotuning. Our evaluation shows good results for the resulting inference performance and resource efficiency.

Contents

Acronyms

VDC vehicle dynamics control

List of Figures

List of Tables

List of Source Codes

1 Introduction

Race cars have fascinated people, built quickly after first car

instead of comfort, max performance

more examples of tradeoffs

1.1 Problem

more than xx years since first car

while mindset is same, now there are electric race cars

new possibilities because of 4wd and computing power

computer assist driver in getting max performance

TC, TV, battery management

components need estimate of vehicle state

task of state estimation which delivers good estimate even in face of sensor failures and unpredictable environment

1.2 Scope

this thesis describes design of a robust, accurate, flexible state estimation for a formula student race car

maybe a research question?

this state estimation fuses available sensors and detects outliers

first background, then design and implementation

then evaluation with measurement data

project for DHBW engineering team but works in other cars as well

2 Background

2.1 Vehicle Dynamics

Vehicle dynamics, basic models

2.2 Estimation Algorithms

Estimation algorithms EFK, filters

Define residual

2.3 Outlier Detection

outlier detection

outlier types: persistent, transient

transient is easier to detect

separate mechanisms

causes: missing connection, no feature correlation for sfii, electromagnetic interference

3 Design

state estimation to be designed will be deployed in two cars

non-autonomous EV, newly built -> sensors can be chosen

partly autonomous DV, reuse old car with camera/lidar -> fixed set of sensors

3.1 Vehicle Characteristics

car is equipped with a plethora of sensors, but only some are important to us

top down view of car with sensor positions

table: sensor name, sensor type, measured variables, columns EV and dv with X

refer to characterization in TC paper

ES910 as ECU, can be programmed in Matlab/Simulink, C, ASCET-MD [1, p. 17]

runs vehicle dynamics control (VDC) with TC, TV, motor request...

The VDC is a tool that helps the driver to exploit the maximum physical potential of the vehicle.

state estimation will be part of VDC

In DV vehicle, DV software runs on a separate computer that needs part of state as well

inputs from sensor via CAN at different frequencies to avoid congestion

it is unclear whether sfii will be in DV

3.2 Requirements

goal: provide robust, accurate estimate of vehicle state for VDC while supporting flexible sensor setups and detecting sensor failures

state variables to estimate

real-time: needs to be computable in 1 ms frequency

complete in all areas, so no developments in near future are necessary

3.3 Architecture

At high level: state estimation = preprocessing + outlier detection + state estimation

To fulfill requirements of robustness and accuracy: outlier detection and sensor fusion

For practical reasons: preprocessing

For flexibility: outlier detection and sensor setup detection using same mechanism

in both cases, sensor cannot or should not be used in sensor fusion

does not matter if not connected, not sending, invalid values

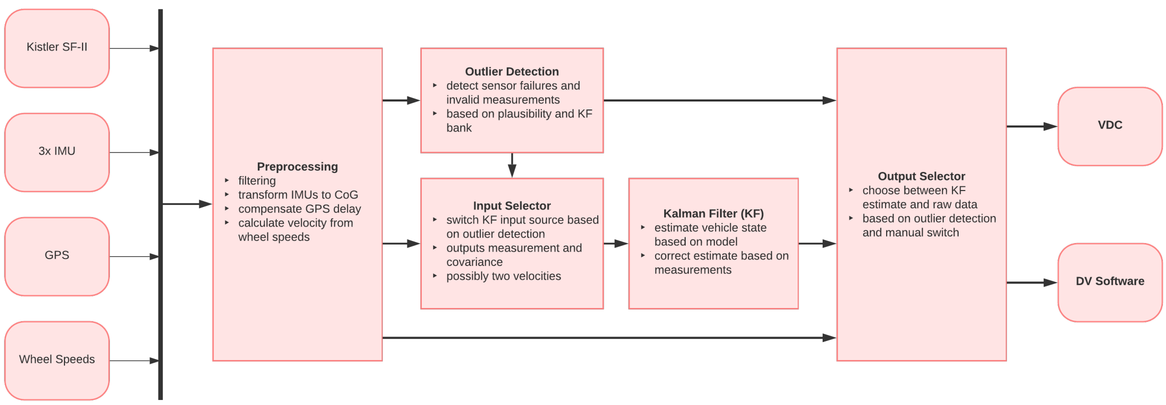

Figure 1: High-level architecture

design principle: use simple methods if they work just as well (occam's razor)

easier to understand and troubleshoot

make less assumptions and thus generalize better

only if the results are not adequate, try more complicated approaches

3.4 Preprocessing

convert to SI units so formulas can be applied

show necessary transformations for each measurement

IMU fusion

3.5 Outlier Detection

Show diagram of AND and OR

3.6 EKF

Show input selection and state equations, jacobians

euler forward discretization

4 Implementation

Since whole VDC is in Simulink because supported by ES910, state estimation as well

Describe Simulink features: subsystems, busses, blocks

Simulink acts as glue, great for modelling signal flow

Matlab code for key algorithm implementations, more readable and unit testable

State estimation is separate model to allow for independent development

Architecture components as subsystems

Reused components referenced model

Strict busses and datatypes, units in simulink to avoid logical errors that are hard to find at compile-time

Integration in VDC

Build process and toolchain incl intecrio and inca for live telematics, monitoring and parametrization

requires application for covariances during testing

5 Evaluation

Fundamental: no ground truth, therefore residuals are a good indicator

non-zero-mean residual means model mismatch [2, p. 158]

Results

Discussion, impact of results on esleek

6 Conclusion

Summary

future work

Bibliography

[1] ETAS GmbH Stuttgart, "ES910.3-A Prototyping and Interface Module User's Guide," 2018. [Online]. Available: `https://www.etas.com/download-center-files/products_ES900/ES910.3-A_UG_R09_EN.pdf`.

[2] Alexander Wischnewski, Tim Stahl, Johannes Betz, and Boris Lohmann, "Vehicle Dynamics State Estimation and Localization for High Performance Race Cars," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 154–161, 2019, ISSN: 2405-8963. DOI: `10.1016/j.ifacol.2019.08.064`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S2405896319303957`.