A real-time state estimation for an electric race car

Student 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

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

CG	center of gravity
GNSS	global navigation satellite system
VDC	vehicle dynamics control

List of Figures

List of Tables

List of Source Codes

List of Symbols

α	Angular accleration	$\mathrm{rad\,s^{-2}}$
$\mathbb{0}$	Zero vector	1
ω	Angular velocity	$\mathrm{rad\,s^{-1}}$
ϕ	Angular displacement around x-axis/roll angle	rad
ψ	Angular displacement around z-axis/yaw angle	rad
θ	Angular displacement around y-axis/pitch angle	rad
φ	Angular orientation	rad
a	Linear accleration	$\mathrm{m\,s^{-2}}$
p	Linear displacement/position in earth-fixed coordinates	m
r	Linear displacement/position in vehicle coordinates	m
v	Linear velocity	$\mathrm{m\,s^{-1}}$
x	x-axis component of linear position in earth-fixed coordinates	m
y	y-axis component of linear position in earth-fixed coordinates	m
z	z-axis component of linear position in earth-fixed coordinates	m

1 Introduction

Race cars have fascinated people, built quickly after first car

instead of comfort, max performance

more examples of tradeoffs

target device

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

individual torque on each wheel

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

The state estimation of a vehicle sits at the intersection of vehicle dynamics and control theory. Both knowledge of the dynamics of a race car and the algorithms to model their physics in equations and software is required to design a successful solution. Therefore, we explore the fundamentals of vehicle dynamics, estimation algorithms and sensor failure detection in this chapter.

Throughout this thesis, the conventions of ISO 8855 [1] will be used, which assumes a right-handed coordinate system. The vehicle coordinate system uses an upward z-axis with a forward x-axis and a leftward y-axis, while the earth-fixed coordinate system uses an upward z-axis with an eastward x-axis and a northward y-axis. The vehicle'scenter of gravity (CG) is used as origin/reference point of the vehicle coordinate system. In case of mixed coordinate systems, left-superscript will be used to denote the reference frame (e.g., ${}^{V}x$ for vehicle coordinates, ${}^{E}x$ for earth-fixed coordinates).

2.1 Rigid Body Kinematics

The fundamental laws of mechanics apply to race cars as they do to any other body. These laws relate, among others, the body's linear and angular position and its time derivatives, resulting in translational and rotational changes. Their three-dimensional vector definitions are shown in equations 1 and 2. To simplify the equations, a rigid body is assumed. This means, that deformations which occur in the vehicle during dynamic maneuvers are negligibly small or vanish. Therefore, points on the body maintain the same distance relative to each other at all times. Furthermore, a two-dimensional motion in the road plane can be assumed in many cases because the effects vertical dynamics are negligible. The simplified equations for that case will also be shown.

$$p = \left[x, y, z\right]^T \tag{1a}$$
$$v = \left[v_x, v_y, v_z\right]^T \tag{1b}$$
$$a = \left[a_x, a_y, a_z\right]^T \tag{1c}$$

$$\varphi = \left[\phi, \theta, \psi\right]^T \tag{2a}$$
$$\omega = \left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^T \tag{2b}$$
$$\alpha = \left[\ddot{\phi}, \ddot{\theta}, \ddot{\psi}\right]^T \tag{2c}$$

The linear displacement p of a body describes its position relative to the origin of its reference frame. It comprises a longitudinal component along the x-axis, a lateral component along the y-axis and a vertical component along the z-axis. Its first time derivative $v = \frac{dr}{dt}$ and second time derivative $a = \frac{d^2r}{dt^2}$ are the body's linear velocity and acceleration, respectively.
The angular orientation φ of a body describes its rotation in the reference frame. It can be described by the roll angle ϕ around the x-axis, the pitch angle θ around the y-axis and the yaw angle ψ around the z-axis. The angular velocity ω and acceleration α describe the element-wise time derivatives of these angles.

2.1.1 Transformation of Linear Velocities

All points on a rigid body experience the same angular velocity, i.e. $\omega^A = \omega^B$ for any two points A, B on that body at all times. However, these two points will generally not have the same linear velocity vector, since it is affected by their location r relative to the CG. Only when $\omega = \mathbb{0}$, the linear velocity is equal for all points on the body, i.e. $v^A = v^B$. If there is a non-zero angular velocity, equation 3 holds for any point P, assuming the velocity at the CG is known (the expanded form can be found in Appendix B.1).

$$v^P = v^{CG} + \omega \times r^P \tag{3}$$

This becomes easier to visualize when regarding the two-dimensional case, where v_z, $\dot{\phi}$ and $\dot{\theta}$ are disregarded, as shown in equation 4.

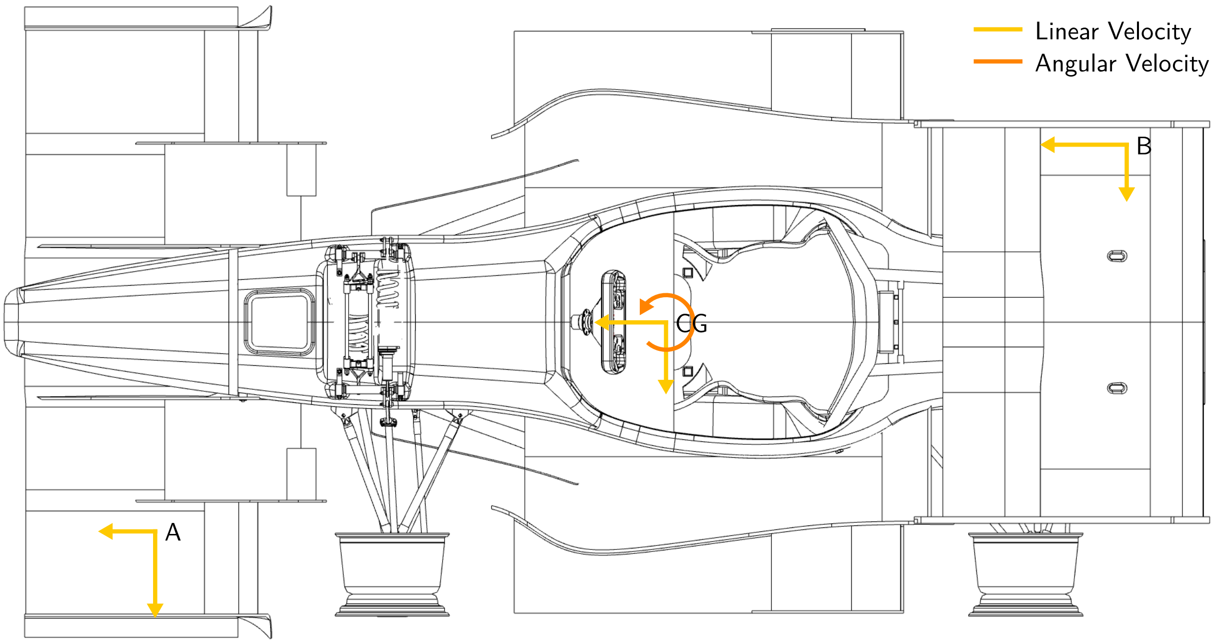

Figure 1: Experienced velocities at off-center points

$$v^P = \begin{bmatrix} v_x^{CG} \\ v_y^{CG} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} v_x^{CG} \\ v_y^{CG} \\ 0 \end{bmatrix} + \begin{bmatrix} -\dot{\psi} \cdot r_y \\ \dot{\psi} \cdot r_x \\ 0 \end{bmatrix} \tag{4}$$

Let us regard the example scenario shown in figure 1, where the vehicle has a positive yaw rate and the CG is moving forward and to the left. Point A is at the front left ($r_x > 0$, $r_y > 0$), while point B is at the rear right ($r_x < 0$, $r_y < 0$). Since r is defined relative to the CG, its position is $\mathbb{0}$. Due to the positive yaw rate, A experiences a higher v_y but lower v_x than the CG. B, on the other hand, experiences a higher v_x but lower v_y than the CG. If the yaw rate were zero, all points would experience the same linear velocity.

2.1.2 Transformation of Linear Accelerations

Like the angular velocity, the angular acceleration is the same at every point on a rigid body, i.e. $\alpha^A = \alpha^B$ for any two points A, B on that body. However, these two points will generally not experience the same linear acceleration. Only if the rotational motion components ω and α are $\mathbb{0}$, the linear acceleration is equal for all points on the body, i.e. $a^A = a^B$. In the general case, equation 5 holds for any point P at location r (the expanded form can be found in Appendix B.2).

$$a^P = a^{CG} + \alpha \times r^P + \omega \times (\omega \times r^P) \tag{5}$$

We can identify two additional components which affect the experienced linear acceleration. The term $\alpha \times r$ is the tangential acceleration along the circular path around the center of rotation, which is $a_{tan} = \alpha \cdot r$ in scalar notation. More significant due to the squared angular velocity, however, is the introduction of the centripetal acceleration a_c in the term $\omega \times (\omega \times r)$. This term is the vector-equivalent of the acceleration resulting from the centripetal force F_c in equation 6.

$$F_c = \frac{mv^2}{r} \iff a_c = \frac{v^2}{r} = \omega^2 r \tag{6}$$

The two-dimensional form, shown in equation 7, is more intuitive than its three-dimensional counterpart. Here, a_z, $\dot{\phi}$, $\dot{\theta}$, $\ddot{\phi}$ and $\ddot{\theta}$ are assumed to be zero. The inward-direction of the centripetal effect can be seen by the negative signs of the angular velocity part.

$$a^P = \begin{bmatrix} a_x^{CG} \\ a_y^{CG} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \ddot{\psi} \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \right) = \begin{bmatrix} a_x^{CG} \\ a_y^{CG} \\ 0 \end{bmatrix} + \begin{bmatrix} -\ddot{\psi} \cdot r_y \\ \ddot{\psi} \cdot r_x \\ 0 \end{bmatrix} + \begin{bmatrix} -\dot{\psi}^2 \cdot r_x \\ -\dot{\psi}^2 \cdot r_y \\ 0 \end{bmatrix} \tag{7}$$

We demonstrate these effects in the example scenario shown in figure 2, which is similar to the one in the previous subsection, but with an additional positive yaw acceleration. In point A, the tangential and centripetal acceleration cancel out and even surpass the linear acceleration experienced in the CG, resulting in a negative a_x and a much lower a_y. The opposite effect occurs in point B, where both a_x and a_y are amplified.

2.1.3 Calculate Angular Acceleration from Linear Acceleration

Direct measurement of the angular acceleration is often not possible. However, individual components of α can be calculated with two known linear acceleration vectors from different points A, B with known locations r^A, r^B using equation 8. It is derived from equation 5, with $\Delta a = a^A - a^B$, $\Delta r = r^A - r^B$ and a^{CG} being eliminated by the difference. The points must not be on the rotation axis of the calculated component of α, otherwise they experience no angular acceleration. Thus, at least three non-collinear points are required to determine all components of α. While the inverse of a cross product is not uniquely determined as can be seen by the scalar factor t, $t = 0$ works well in practice.

$$\Delta a = \alpha \times \Delta r + \omega \times (\omega \times \Delta r) \implies \alpha = \frac{\Delta r \times (\Delta a - \omega \times (\omega \times \Delta r))}{\|\Delta r\|^2} + t \cdot \Delta r, t \in \mathbb{R} \tag{8}$$

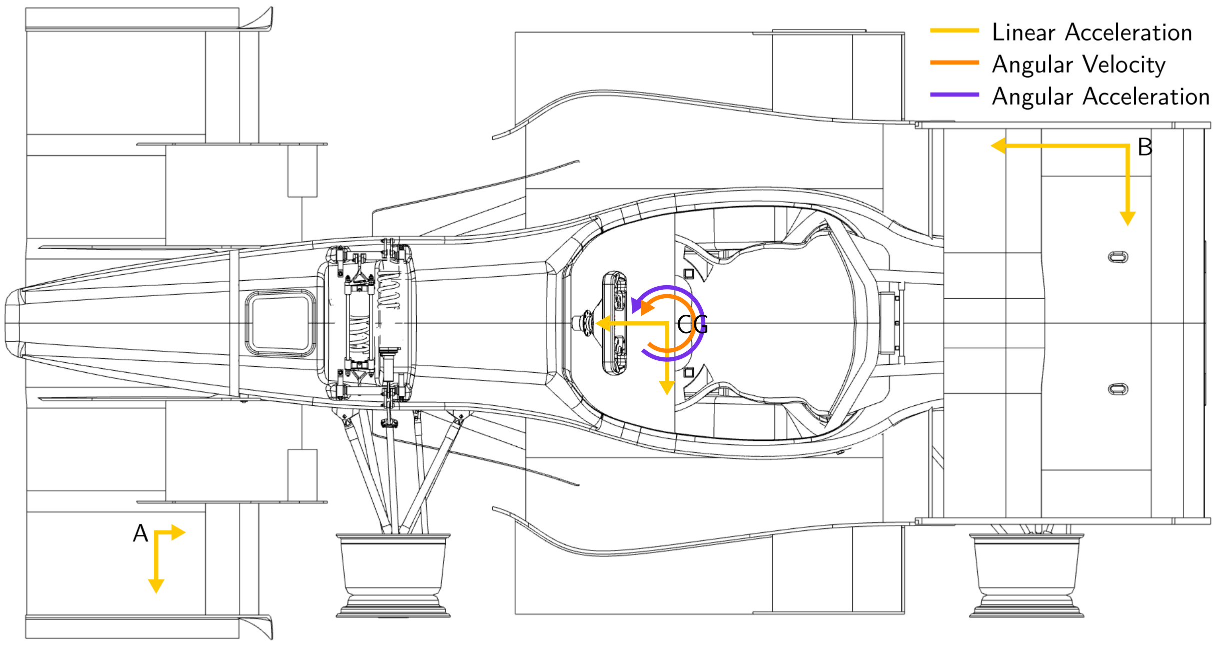

Figure 2: Experienced acceleration at off-center points

In two dimensions, this becomes easier to understand. Calculation of the yaw acceleration from the longitudinal and lateral accelerations of two points is shown in equation 9. We can see that the accelerations and distances from different axes are cross-correlated. The equation even shows how positioning both points on the z-axis would result in a zero division, which shows that points off the rotation axis are required. When A and B are directly in front of and behind the CG on the x-axis, the equation simplifies to $\ddot{\psi} = \frac{\Delta a_y}{\Delta r_x}$. To minimize effects of measurement uncertainty, the two points should be located

$$\alpha = \frac{\begin{bmatrix}\Delta r_x \\ \Delta r_y \\ 0\end{bmatrix} \times \left(\begin{bmatrix}\Delta a_x \\ \Delta a_y \\ 0\end{bmatrix} - \begin{bmatrix}-\dot{\psi}^2 \cdot r_x \\ -\dot{\psi}^2 \cdot r_y \\ 0\end{bmatrix} \right)}{\left\| \begin{matrix}\Delta r_x \\ \Delta r_y \\ 0\end{matrix} \right\|^2} \implies = \ddot{\psi} = \frac{\Delta r_x \Delta a_y - \Delta r_y \Delta a_x}{\Delta r_x^2 + \Delta r_y^2} \tag{9}$$

2.1.4 Transformation Between Coordinate Systems

Most sensor measurements are done in the vehicle coordinate system. Others, such as global navigation satellite system (GNSS) (e.g., Global Positioning System (GPS), Galileo, GLONASS) measurements, will be made in earth-fixed coordinates, however. Therefore, we need to transform the quantities from equations 1 and 2 from the earth-fixed coordinate system to the vehicle coordinate system, especially if we want to relate these measurements.

Equations 10 and 11 show these transformations for a non-moving, non-accelerating earth-fixed reference frame.

$$
\begin{align}
{}^{V}p &= {}^{E}O + {}^{E}p \tag{10a}\\
{}^{V}v &= R \cdot {}^{E}v \tag{10b}\\
{}^{V}a &= R \cdot {}^{E}a \tag{10c}
\end{align}
$$

$$
\begin{align}
\varphi &= \left[\phi, \theta, \psi\right]^{T} \tag{11a}\\
\omega &= \left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^{T} \tag{11b}\\
\alpha &= \left[\ddot{\phi}, \ddot{\theta}, \ddot{\psi}\right]^{T} \tag{11c}
\end{align}
$$

motion equations

wheels to speed

2.2 Estimation Algorithms

goal: solve filtering problem and fuse sensors to obtain optimal state estimate

http://www.anuncommonlab.com/articles/how-kalman-filters-work/index.html

Estimation algorithms

particle filter

kalman filter

EFK

compare

Define residual

disable measurements using separate mask

2.3 Failure Detection

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

outlier types: persistent, transient [2, p. 170 ff.]

outlier, drift, null [3, p. 19 f.]

transient is easier to detect

separate mechanisms

statistical methods

transient: range and

EKF bank

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

e.g., sensors for brake pressure or accelerator pedal actuation not relevant

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

sfii is key sensor for velocity estimation because it is slip free as opposed to wheels and fast as opposed to GPS

(compare velocity with only gps, sfii and whee)

ES910 as ECU, can be programmed in Matlab/Simulink, C, ASCET-MD [4, 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 and DV software

flexible: support arbitrary sensor setups within full sensor setup

robust: detect and handle sensor failures

state variables to estimate

estimate these variables in CoG, this describes motion of vehicle since it is assumed to be a rigid body

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

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

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

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

easier to understand and troubleshoot

make less assumptions and thus generalize better

because maintainers change every year

on a higher level, clear architecture to facilitate maintenance and extension

We assume that vertical dynamics are negligible and only motion in the two-dimensional plane of the track are relevant, simplifying equations.

3.3 Architecture

in the literature, there is no concrete solution for our problem

however, we can combine common components into our own solution

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

preprocessing and outlier detection for each sensor in parallel

all IMUs have same treatment

out bus creation calculates vehicle side slip angle from vx and vy

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

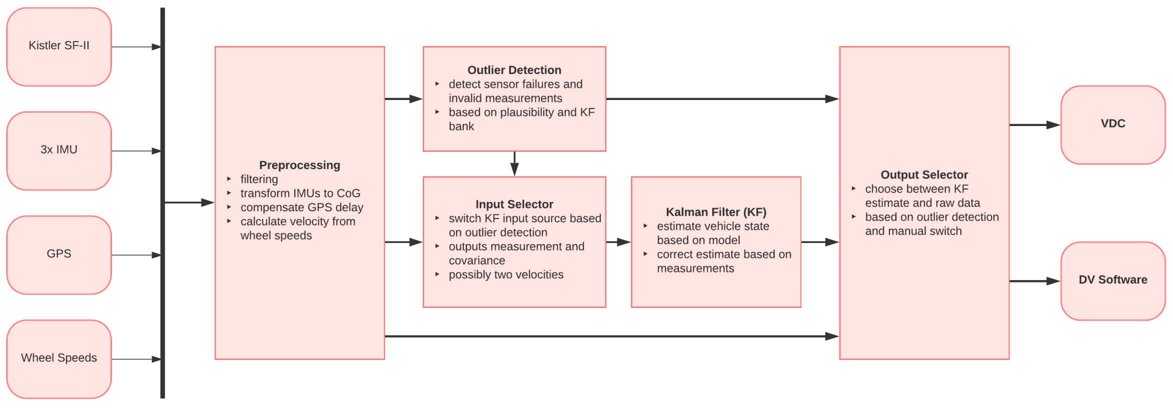
Kistler SF-II
3x IMU
GPS
Wheel Speeds
Preprocessing
filtering
transform IMUs to CoG
compensate GPS delay
calculate velocity from wheel speeds
Outlier Detection
detect sensor failures and invalid measurements
based on plausibility and KF bank
Input Selector
switch KF input source based on outlier detection
outputs measurement and covariance
possibly two velocities
Kalman Filter (KF)
estimate vehicle state based on model
correct estimate based on measurements
Output Selector
choose between KF estimate and raw data
based on outlier detection and manual switch
VDC
DV Software

Figure 3: High-level architecture

Show all inputs and outputs

3.4 Preprocessing

convert to SI units and deg to rad so formulas can be applied

show necessary transformations for each measurement

IMU fusion

very important because used extensively in EKF and other parts of VDC like TC and TV (especially dpsi)

fast response

contains outlier detection because available sensors need to be known before fusion in preprocessing

GPS WGS84 coordinates transformed to track with first known point as origin

sfii transformed to CG

3.5 Outlier Detection

Show diagram of AND and OR

For all EKF inputs

Plausibility check for all

For velocity, use wheels, GPS and sfii in EKF bank

Plausibility for velocities rather conservative

debouncing to increase robustness

allow manual three-way override to enable/disable sensors and override outlier detection errors

3.6 EKF

outlier detection sensor state is used to create measurement mask

Show input selection and state equations, jacobians

euler forward discretization

GPS is not included because it is not beneficial due to its slow response

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, clear architecture requirement

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

only model-in-the-loop testing with measurement data because complete setup was not available during development for HIL

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 [5, p. 158]

Results

bad position in EV without heading measurements, because velocity cannot be used properly to estimate position

compare raw measurements with velocity estimate

show results for each state variable

effect of different debouncing times and thresholds

max likelihood IMU fusion not better for omega and a, but less noise for alpha when more than 2 imus -> better EKF results

Mean approach works well when all IMUs measure the same components of alpha

Discussion, impact of results on esleek

prepare for DV

6 Conclusion

Summary

future work

Bibliography

[1] ISO, *Road vehicles — Vehicle dynamics and road-holding ability — Vocabulary*, 2011.

[2] H. Himmelblau, J. H. Wise, A. G. Piersol, and M. R. Grundvig, *Handbook for Dynamic Data Acquisition and Analysis - IES Recommended Practices 012.1.* 1994. [Online]. Available: `https://trs.jpl.nasa.gov/bitstream/handle/2014/33780/94-0509.pdf`.

[3] J. Kabzan, M. d. I. La Valls, V. Reijgwart, H. F. C. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan, A. Dhall, E. Chisari, N. Karnchanachari, S. Brits, M. Dangel, I. Sa, R. Dubé, A. Gawel, M. Pfeiffer, A. Liniger, J. Lygeros, and R. Siegwart, *AMZ Driverless: The Full Autonomous Racing System.* [Online]. Available: `https://arxiv.org/pdf/1905.05150.pdf`.

[4] 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`.

[5] 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`.

Glossary

target device

the device that inference will be performed on; usually an accelerator located at the edge

A Bus Definitions

Bus definitions

B Expanded Rigid Body Equations

B.1 Transformation of Linear Velocity

$$\begin{aligned}
\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} &= \begin{bmatrix} v_x^{CG} \\ v_y^{CG} \\ v_z^{CG} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \\
&= \begin{bmatrix} v_x^{CG} \\ v_y^{CG} \\ v_z^{CG} \end{bmatrix} + \begin{bmatrix} \dot{\theta} r_z - \dot{\psi} r_y \\ \dot{\psi} r_x - \dot{\phi} r_z \\ \dot{\phi} r_y - \dot{\theta} r_x \end{bmatrix}
\end{aligned}$$

B.2 Transformation of Linear Acceleration

$$\begin{aligned}
\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} &= \begin{bmatrix} a_x^{CG} \\ a_y^{CG} \\ a_z^{CG} \end{bmatrix} + \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \left(\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \right) \\
&= \begin{bmatrix} a_x^{CG} \\ a_y^{CG} \\ a_z^{CG} \end{bmatrix} + \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} \dot{\theta} r_z - \dot{\psi} r_y \\ \dot{\psi} r_x - \dot{\phi} r_z \\ \dot{\phi} r_y - \dot{\theta} r_x \end{bmatrix} \\
&= \begin{bmatrix} a_x^{CG} \\ a_y^{CG} \\ a_z^{CG} \end{bmatrix} + \begin{bmatrix} \ddot{\theta} r_z - \ddot{\psi} r_y \\ \ddot{\psi} r_x - \ddot{\phi} r_z \\ \ddot{\phi} r_y - \ddot{\theta} r_x \end{bmatrix} + \begin{bmatrix} \dot{\theta}(\dot{\phi} r_y - \dot{\theta} r_x) - \dot{\psi}(\dot{\psi} r_x - \dot{\phi} r_z) \\ \dot{\psi}(\dot{\theta} r_z - \dot{\psi} r_y) - \dot{\phi}(\dot{\phi} r_y - \dot{\theta} r_x) \\ \dot{\phi}(\dot{\psi} r_x - \dot{\phi} r_z) - \dot{\theta}(\dot{\theta} r_z - \dot{\psi} r_y) \end{bmatrix}
\end{aligned}$$