Implementation in embedded systems of state observers based on multibody dynamics

Antonio J. Rodríguez

Doctoral thesis

Ferrol, June 25th, 2020

UNIVERSIDADE DA CORUÑA

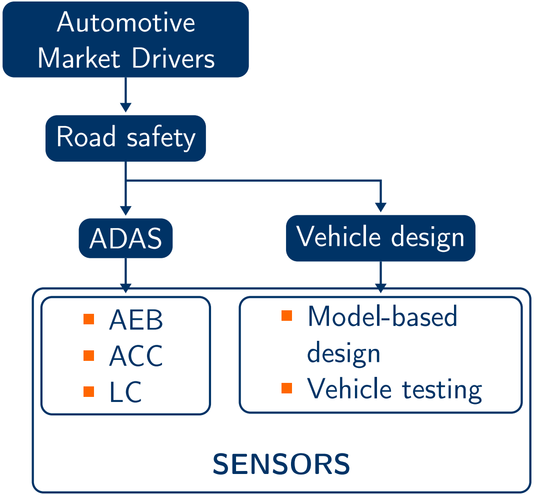
Outline

1. Introduction
2. Model-based state observers
3. New generation embedded hardware
4. Use-case application
5. Conclusions and future work

Outline

1. Introduction
2. Model-based state observers
3. New generation embedded hardware
4. Use-case application
5. Conclusions and future work

Motivation


Automotive Market Drivers
Road safety
ADAS
Vehicle design
AEB
ACC
LC
Model-based design
Vehicle testing
SENSORS

AC R 2015

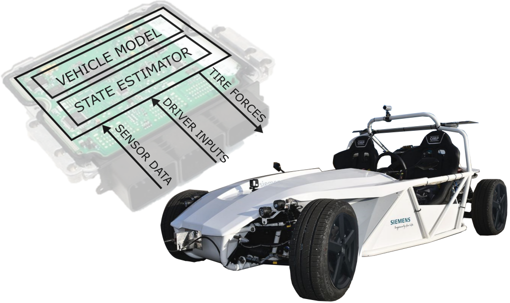
Use-Case: Wheel Force Transducers

Real sensors

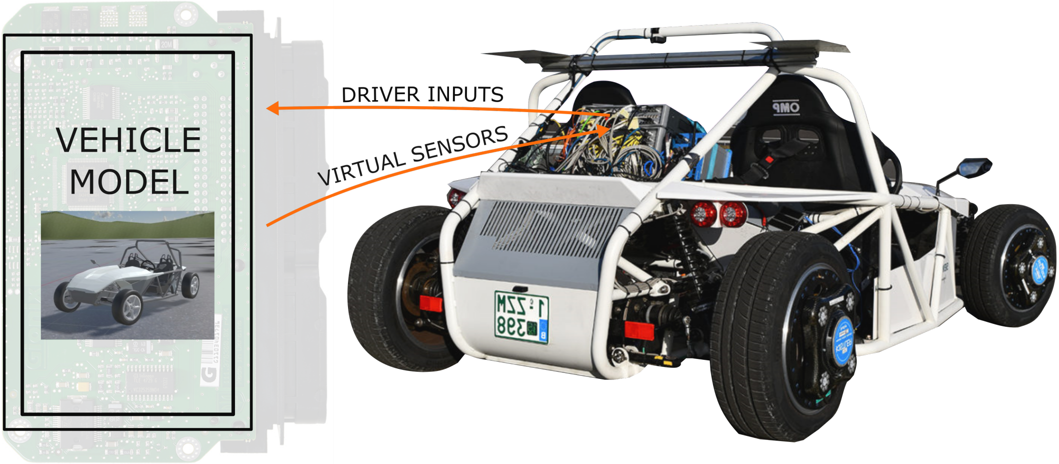
- Instrumented rim
- Based on strain gauges
- Expensive

Virtual sensors

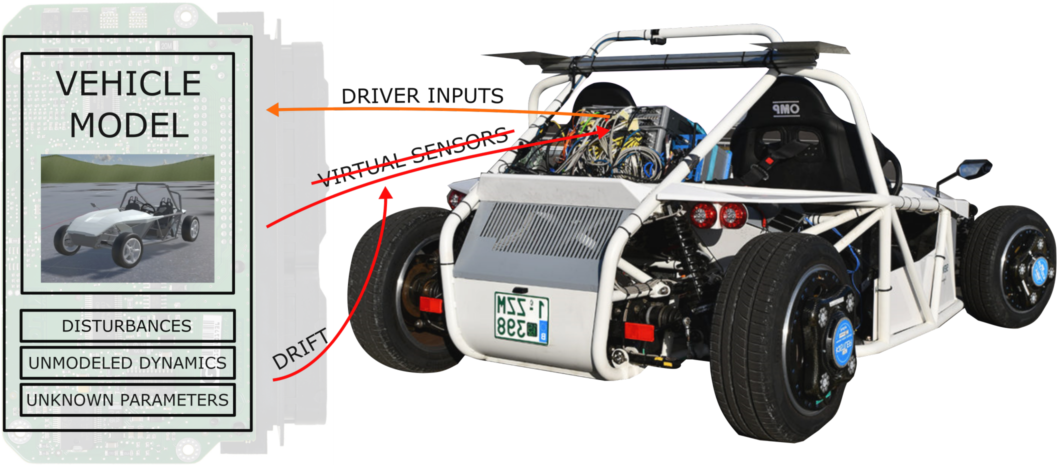
- Model-based
- Virtual enviroment
- Minimal set of sensors



Virtual Sensing



Virtual Sensing



VEHICLE MODEL

DISTURBANCES

UNMODELED DYNAMICS

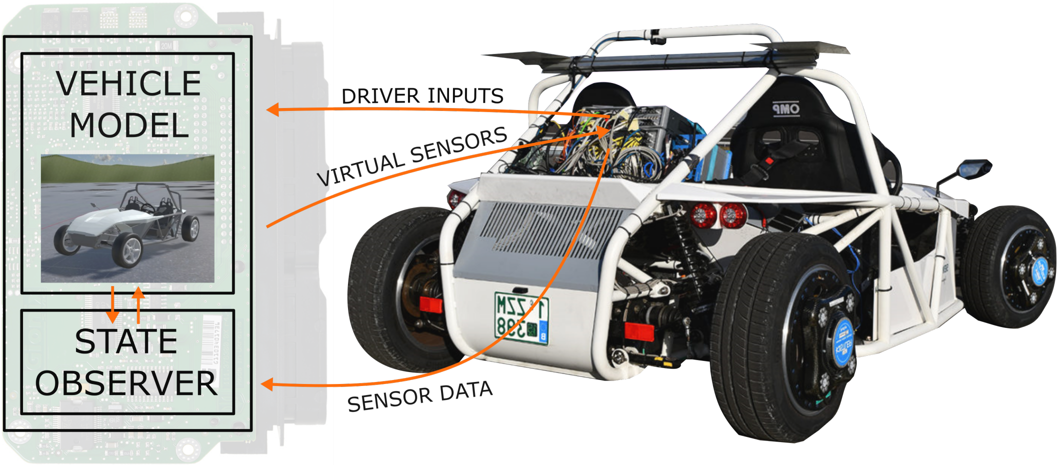
UNKNOWN PARAMETERS

DRIVER INPUTS

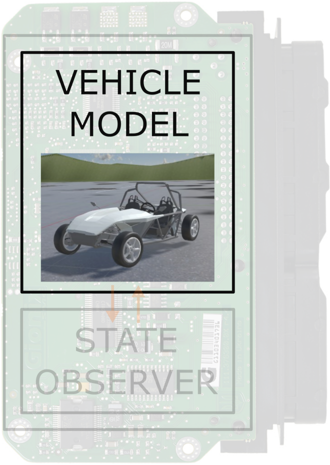
VIRTUAL SENSORS

DRIFT

Virtual Sensing



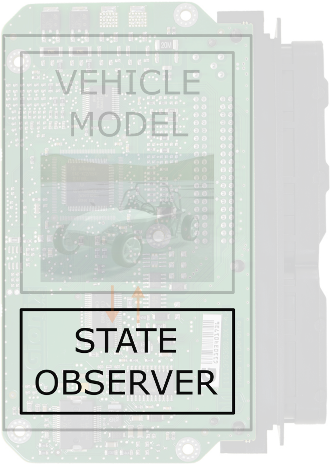
Virtual Sensing: Vehicle modeling



Vehicle Model

Analytical Models	Multibody Models
✓ Low computational cost	✗ High computational cost
✓ Reduced complexity	✓ More virtual sensors
✓ Less vehicle parameters	✗ More vehicle parameters
✗ Lower accuracy	✓ Higher accuracy
✗ Maneuver specific	✓ More versatile

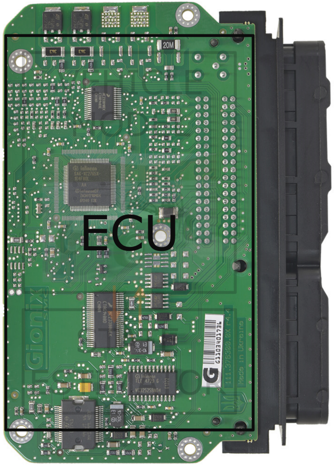
Virtual Sensing: State observer



State observer

- Several options:
  - Particles filter
  - Moving horizon estimator
  - Kalman filter
  - etc
- Kalman filter is widely used in automotive applications
- Strong background in the LIM
  - Evaluation of KF with MB models
  - New efficient KF developed

Embedded hardware: Electronic Control Unit (ECU)

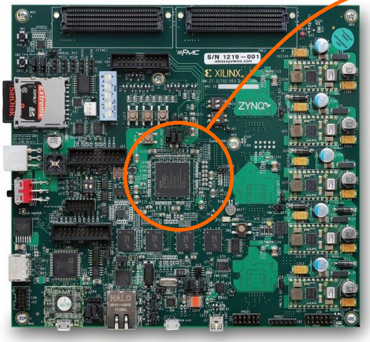


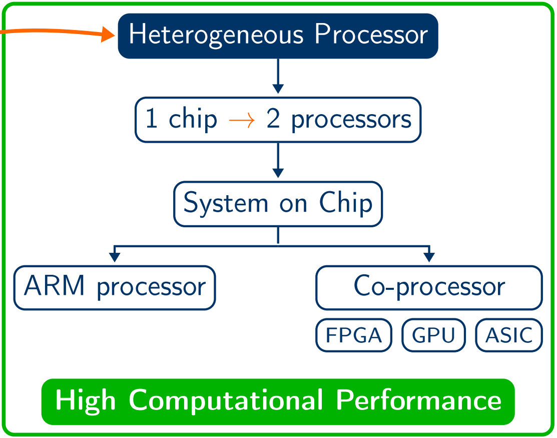
ECU

- In-vehicle embedded hardware
- Automotive standard complient:
  - Reliability
  - Timing
  - Safety
- Low energy consumption
- Low computational capabilities

New generation embedded hardware

New generation ECU





Heterogeneous Processor

1 chip → 2 processors

System on Chip

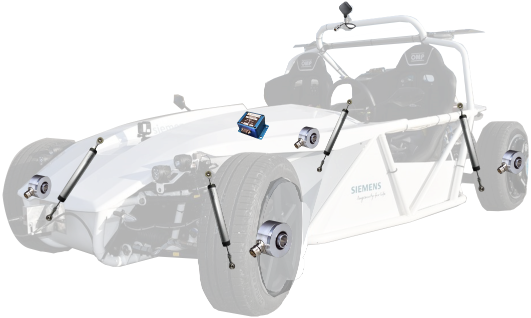
ARM processor

Co-processor

FPGA GPU ASIC

High Computational Performance

Objectives



Virtual sensors	State observer	■ Minimal set of sensors ■ Estimate any variable: tire forces...
	Vehicle MB model	■ Efficient formulation and programming ■ Accurate dynamic simulation
	Real time	New generation embedded hardware

Objectives

Implement accurate virtual sensors for real-time in-vehicle applications

Study the suitability of FPGAs for accelerating MB simulations

Develop an accurate and efficient MB-based state observer for vehicle dynamics

Develop a friendly framework for an easy real implementation of the solution

Outline

1. Introduction
2. Model-based state observers
3. New generation embedded hardware
4. Use-case application
5. Conclusions and future work

Multibody dynamics

MB model

Assembly of two or more bodies imperfectly joined, having the possibility of relative movement between them

MBScoder [1]

Open source library for automatic code generation for MB dynamics

- Efficient code in multiple languages
- Different MB formulations
- Different MB coordinates
  - Natural
  - Relative (added in this thesis)

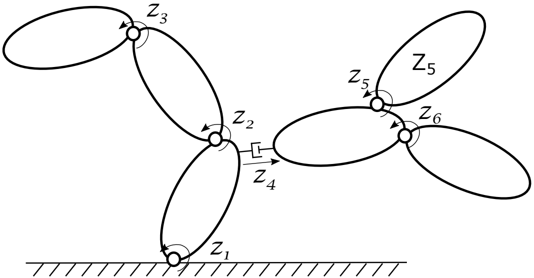
[1] R. Pastorino, F. Cosco, F. Naets, W. Desmet, and J. Cuadrado, "Hard real-time multibody simulations using ARM-based embedded systems," Multibody System Dynamics, vol. 37, pp. 127–143, May 2016.

Semi-recursive method [2]

Relative coordinates

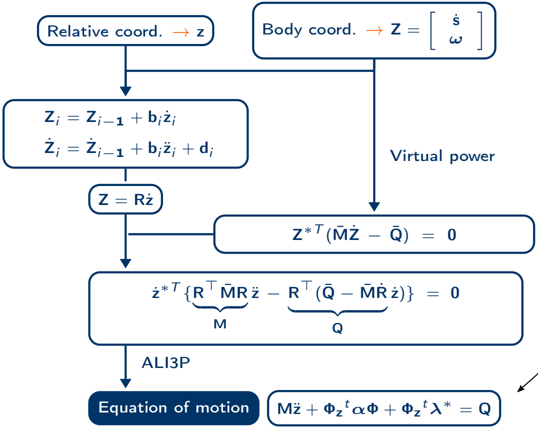
Each body is defined with respect to its previous body

- ✓ Minimum number of variables
- ✗ Complex equation of motion definition



[2] J. Cuadrado, D. Dopico, M. Gonzalez, and M. A. Naya, "A combined penalty and recursive real-time formulation for multibody dynamics," *Journal of Mechanical Design*, vol. 126, no. 4, p. 602, 2004.

Semi-recursive method [2]



Relative coord. → $\mathbf{z}$

Body coord. → $\mathbf{Z} = \begin{bmatrix} \dot{\mathbf{s}} \\ \boldsymbol{\omega} \end{bmatrix}$

$$\mathbf{Z}_i = \mathbf{Z}_{i-1} + \mathbf{b}_i \dot{\mathbf{z}}_i$$

$$\dot{\mathbf{Z}}_i = \dot{\mathbf{Z}}_{i-1} + \mathbf{b}_i \ddot{\mathbf{z}}_i + \mathbf{d}_i$$

$\mathbf{Z} = \mathbf{R}\dot{\mathbf{z}}$

Virtual power

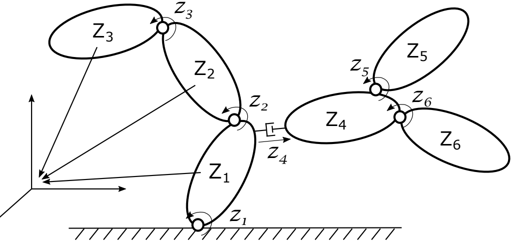
$$\mathbf{Z}^{*T}(\bar{\mathbf{M}}\dot{\mathbf{Z}} - \bar{\mathbf{Q}}) = \mathbf{0}$$

$$\dot{\mathbf{z}}^{*T}\{\underbrace{\mathbf{R}^{\top}\bar{\mathbf{M}}\mathbf{R}}_{\mathbf{M}}\ddot{\mathbf{z}} - \underbrace{\mathbf{R}^{\top}(\bar{\mathbf{Q}} - \bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}})}_{\mathbf{Q}}\} = \mathbf{0}$$

ALI3P

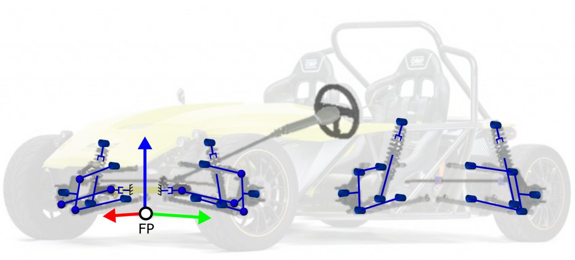
Equation of motion

$$\mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}{}^{t}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}{}^{t}\boldsymbol{\lambda}^{*} = \mathbf{Q}$$



[2] J. Cuadrado, D. Dopico, M. Gonzalez, and M. A. Naya, "A combined penalty and recursive real-time formulation for multibody dynamics," *Journal of Mechanical Design*, vol. 126, no. 4, p. 602, 2004.

Vehicle MB model



Summary	
DOFs ($\mathbf{z}^i$)	14
Steer	Kinematically guided
Rel. Coords. ($\mathbf{z}^d$)	42
Bodies	29
Constraints	42
Tire model	TMeasy [3]

[3] W. Hirschberg, G. Rill, and H. Weinfurter, "Tire model TMeasy," *Vehicle System Dynamics*, vol. 45, pp. 101–119, Jan. 2007.

Kalman filters based on MB dynamics

CEKF

- EKF in continuous form
- Non-linearities approximated by a Jacobian matrix
- Requires to adapt MB equations
- Stability and accuracy problems with low sampling rates

DEKF

- EKF in discrete form
- Non-linearities approximated by a Jacobian matrix
- Requires to adapt MB equations

UKF

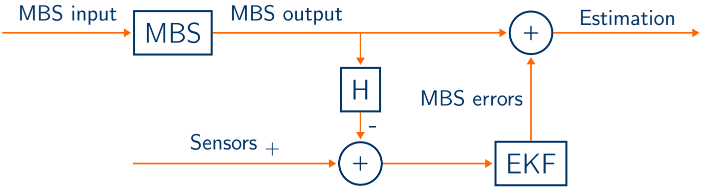
- Based on a set of deterministically chosen weighted sample points (sigma-points)
- Sigma-points are propagated through the MB equations
- Independence from MB equations and KF
- High computational cost

errorEKF [4]

- Indirect Kalman filter
- EKF based on the errors in the MB variables
- Independence from MB equations and KF
- High computational efficiency

[4] E. Sanjurjo, D. Dopico, A. Luaces, and M. A. Naya, "State and force observers based on multibody models and the indirect Kalman filter," *Mechanical Systems and Signal Processing*, vol. 106, pp. 210–228, June 2018.

KF: errorEKF with force estimation



1º: Prediction

$$\mathbf{x} = \left[\Delta\mathbf{z}^i, \Delta\dot{\mathbf{z}}^i, \Delta\ddot{\mathbf{z}}^i\right]$$

2º: Propagation

$$\hat{\mathbf{x}}_k^- = \mathbf{0}$$
$$\mathbf{P}_k^- = \mathbf{f_x}_{k-1}\mathbf{P}_{k-1}^+\mathbf{f_x}_{k-1}^\top + \mathbf{\Sigma}^P$$

3º: KF Correction

$$\tilde{\mathbf{y}}_k = \mathbf{o}_k - \mathbf{h}(\mathbf{z}_k, \dot{\mathbf{z}}_k, \ddot{\mathbf{z}}_k)$$
$$\mathbf{\Sigma}_k = \mathbf{h_x}\mathbf{P}_k^-\mathbf{h_x}^\top + \mathbf{\Sigma}_k^S$$
$$\mathcal{K}_k = \mathbf{P}_k^-\mathbf{h_x}^\top\mathbf{\Sigma}_k^{-1}$$
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathcal{K}_k\tilde{\mathbf{y}}_k$$
$$\mathbf{P}_k^+ = (\mathbf{I} - \mathcal{K}_k\mathbf{h_x})\mathbf{P}_k^-$$

4º: MB Correction

$$\hat{\mathbf{x}}_k^+ \rightarrow \left[\Delta\hat{\mathbf{z}}^d, \Delta\hat{\dot{\mathbf{z}}}^d, \Delta\hat{\ddot{\mathbf{z}}}^d\right]$$
$$\Delta\hat{\mathbf{Q}}^d = \mathbf{0}$$
$$\Delta\hat{\mathbf{Q}}^i = \hat{\mathbf{M}}^i\ddot{\mathbf{z}}^i - \hat{\mathbf{Q}}^i$$
$$\Delta\hat{\mathbf{Q}} = \left[\ \Delta\hat{\mathbf{Q}}^i \quad \Delta\hat{\mathbf{Q}}^d\ \right]$$

Outline

1 Introduction

2 Model-based state observers

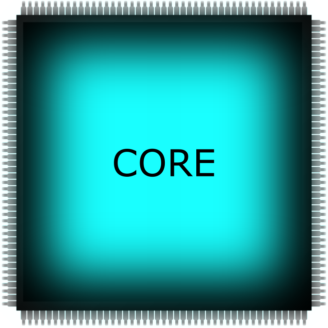
3 New generation embedded hardware

4 Use-case application

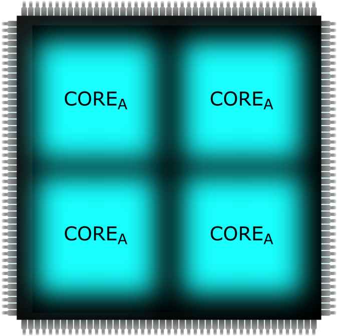
5 Conclusions and future work

Modern hardware analysis

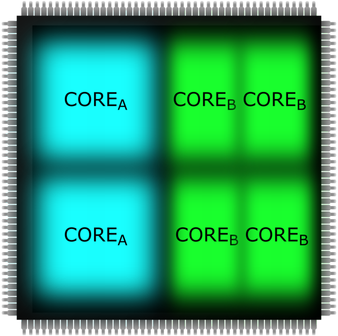
Single Core Processor



Homogeneous Multicore Processor



Heterogeneous Multicore Processor


CORE_A
CORE_B CORE_B
CORE_A
CORE_B CORE_B

Modern hardware analysis

Single Core Processor

- ✓ High computational power in one core
- ✓ Simpler to program
- ✗ Over-provisioned: same core for any application
  - ✗ Low area efficiency
  - ✗ Low power efficiency

Homogeneous Multicore Processor

CORE$_A$
CORE$_A$
CORE$_A$
CORE$_A$

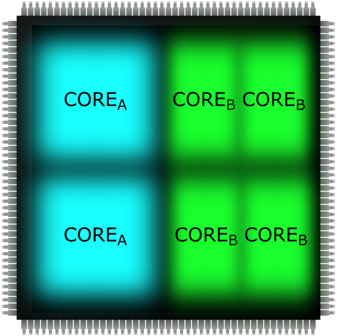
- ✗ Lower computational power per core
- ✓ High computational power through parallelization
  - ✓ Improved area efficiency
  - ✓ Improved power efficiency
- ✗ Over-provisioned

Heterogeneous Multicore Processor

CORE$_A$
CORE$_B$
CORE$_B$
CORE$_A$
CORE$_B$
CORE$_B$

- ✗ Lower computational power per core
- ✓ High computational power through parallelization
- ✓ Fit-for-purpose cores
  - ✓ Highest area efficiency
  - ✓ Highest power efficiency

Heterogeneous processors for scientific computing



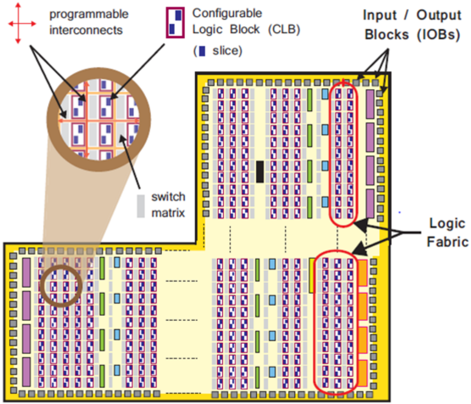
$CORE_A$ — Conventional embedded processor

$CORE_B$ — ASIC
- ✓ Tailored for an application
- ✓ Highest performance
- ✗ Expensive
- ✗ Long development times

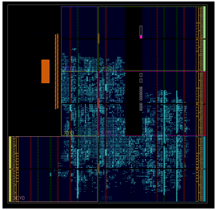
$CORE_B$ — GPU
- ✓ Programmable
- ✓ Large number of cores
- ✗ Best suited for larger systems
- ✗ All cores are of the same type

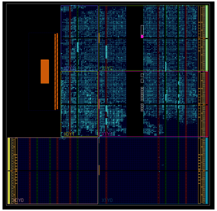
$CORE_B$ — FPGA
- ✓ Programmable
- ✓ Fit-for-purpose
- ✓ Freedom for parallelization
- ✗ Limited hardware resources

Field Programmable Gate Array (FPGA)


programmable interconnects
Configurable Logic Block (CLB)
(slice)
Input / Output Blocks (IOBs)
switch matrix
Logic Fabric

- Set of wires, logic gates and registers
- Combining each element, a dedicated computer unit can be "built" for a specific application
- Freedom in design to improve performance



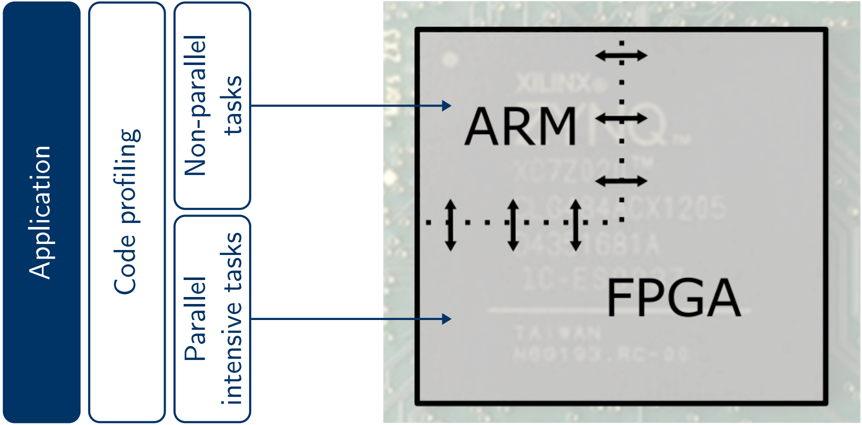


Selected Hardware

Zynq-7000 XC7Z020

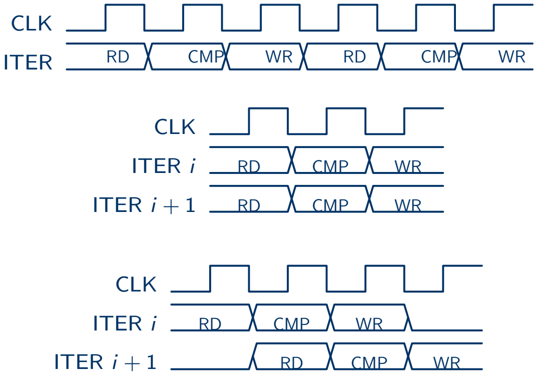
- ARM Cortex-A9
  - Dual core
  - Max. freq: 667 MHz
- FPGA Artix-7
- Low-end device (2012)
- Commonly used in automotive applications
  - Computer vision
  - Control purposes

Hardware/Software partitioning



Parallelization

Parallelization Leads to an increment on the computational efficiency



Remarks

- Different types:
  - Unroll
  - Pipeline
- Data dependency is an important factor

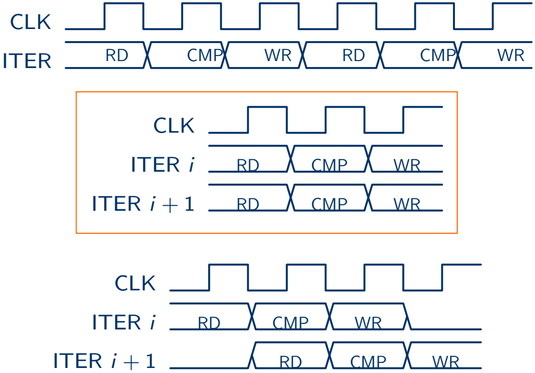
Parallelization

Parallelization

Leads to an increment on the computational efficiency

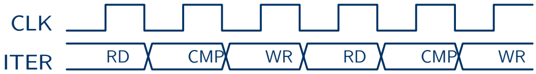
Remarks

- Different types:
  - Unroll
  - Pipeline
- Data dependency is an important factor



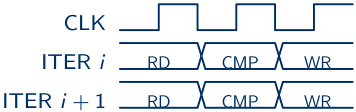
Parallelization

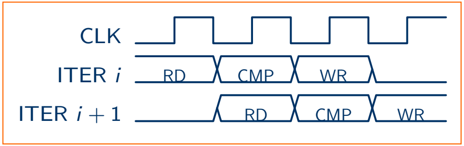
Parallelization Leads to an increment on the computational efficiency



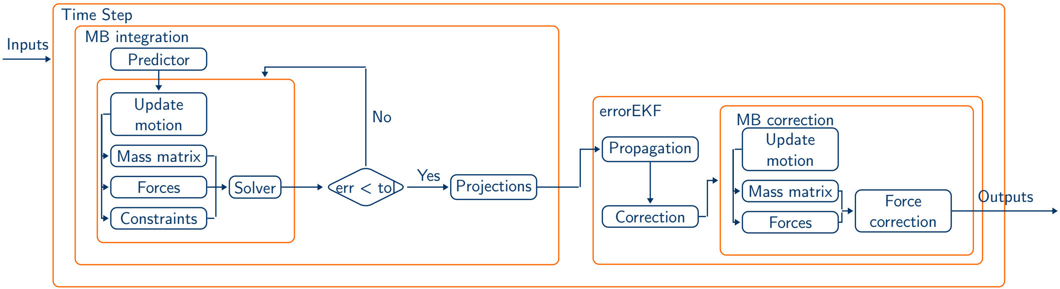
Remarks

- Different types:
  - Unroll
  - Pipeline
- Data dependency is an important factor

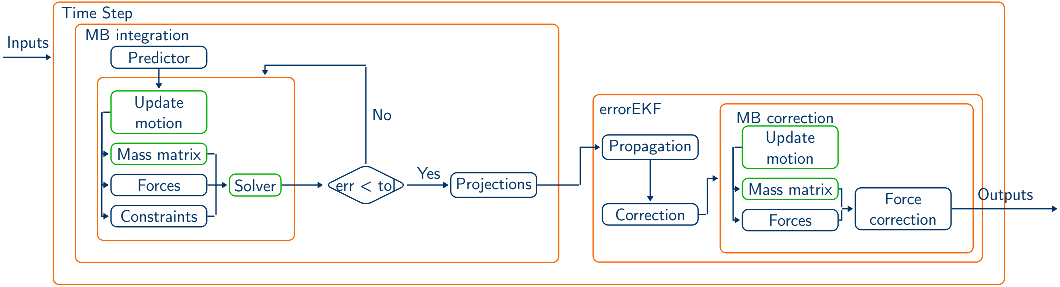




Virtual sensors algorithm: profiling



Virtual sensors algorithm: profiling



Inputs

Time Step

MB integration

Predictor

Update motion

Mass matrix

Forces

Constraints

Solver

err < tol

No

Yes

Projections

errorEKF

Propagation

Correction

MB correction

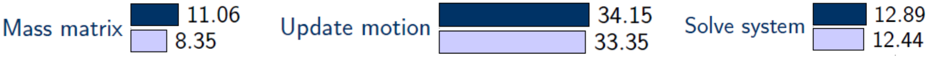
Update motion

Mass matrix

Forces

Force correction

Outputs



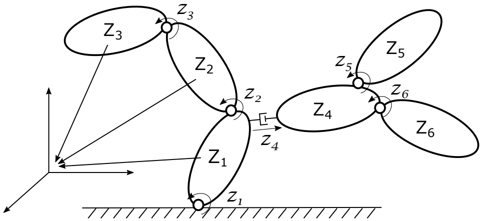
Mass matrix calculation

No data dependency

Body coordinates

Indiv. Mass Matrix

$$\bar{\mathbf{M}}_i = \begin{bmatrix} m\mathbf{I} & -m\tilde{\mathbf{g}} \\ m\tilde{\mathbf{g}} & \mathbf{J} - m\tilde{\mathbf{g}}\tilde{\mathbf{g}} \end{bmatrix}$$



Coordinates relation

$$\mathbf{Z}_i = \mathbf{Z}_{i-1} + \mathbf{b}_i \dot{\mathbf{z}}_i$$

$$\dot{\mathbf{Z}}_i = \dot{\mathbf{Z}}_{i-1} + \mathbf{b}_i \ddot{\mathbf{z}}_i + \mathbf{d}_i$$

$$\mathbf{Z} = \mathbf{R}\dot{\mathbf{z}}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{b}_1 & \mathbf{0} & \mathbf{0} & \mathbf{b}_4 & \mathbf{0} & \mathbf{0} \\ \mathbf{b}_1 & \mathbf{0} & \mathbf{0} & \mathbf{b}_4 & \mathbf{b}_5 & \mathbf{0} \\ \mathbf{b}_1 & \mathbf{0} & \mathbf{0} & \mathbf{b}_4 & \mathbf{0} & \mathbf{b}_6 \end{bmatrix}$$

Data dependency

Relative coordinates

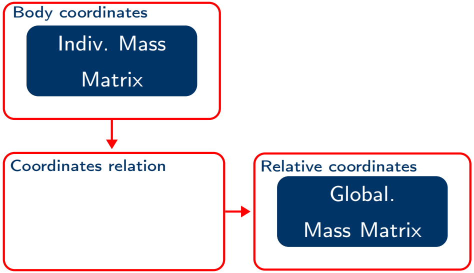
Global. Mass Matrix

$$\mathbf{M} = \mathbf{R}^{\top}\bar{\mathbf{M}}\mathbf{R}$$

FPGA implementation: mass matrix calculation (strategy 1)

Strategy 1		
	ARM	MB-based state observer
	FPGA	Full mass matrix calculation

NOT ENOUGH FPGA RESOURCES



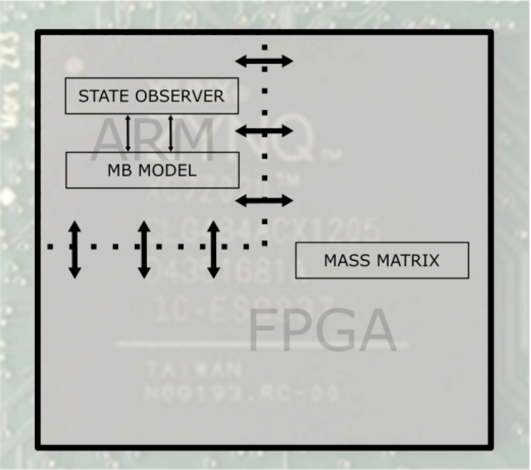
Body coordinates

Indiv. Mass Matrix

Coordinates relation

Relative coordinates

Global. Mass Matrix



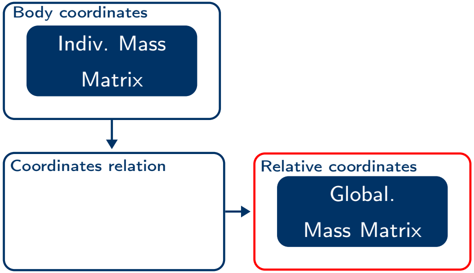
FPGA implementation: mass matrix calculation (strategy 2)

Strategy 2

ARM — MB-based state observer

FPGA — Assembly mass matrix

NOT ENOUGH FPGA RESOURCES



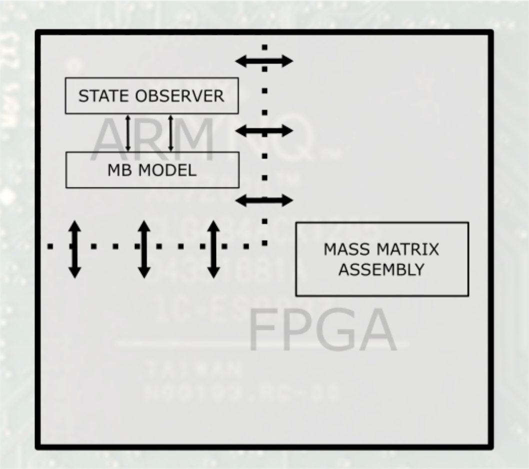
Body coordinates

Indiv. Mass Matrix

Coordinates relation

Relative coordinates

Global. Mass Matrix

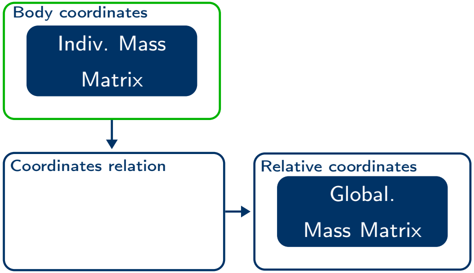


FPGA implementation: mass matrix calculation (strategy 3)

Strategy 3

ARM | MB-based state observer

FPGA | Indiv. mass matrices



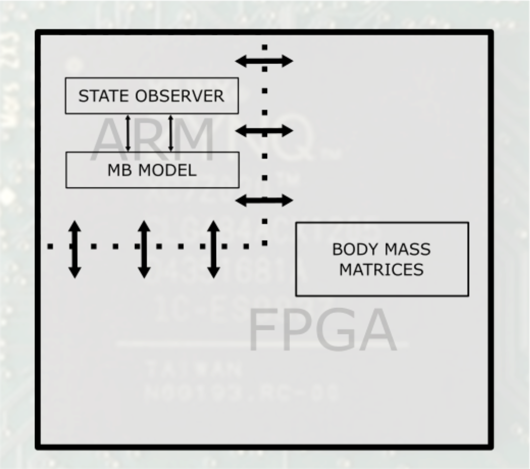
Body coordinates

Indiv. Mass Matrix

Coordinates relation

Relative coordinates

Global. Mass Matrix



FPGA implementation: mass matrix calculation (strategy 3)

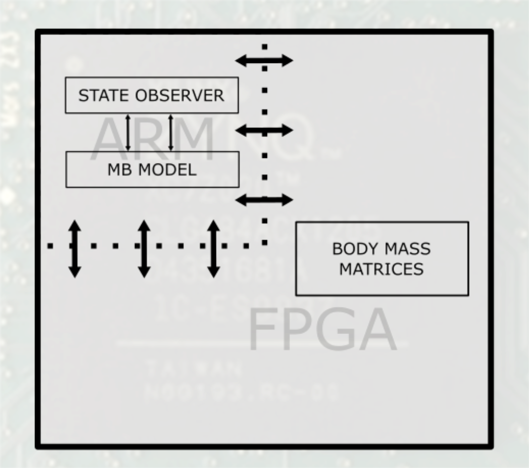
Strategy 3

ARM — MB-based state observer

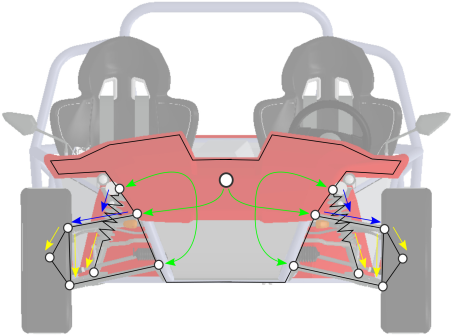
FPGA — Indiv. mass matrices

Implementation summary

Parallel	Resources	Latency	Speed-up
No	Yes	696	-
Unroll	No	24	x29
Pipeline	Yes	52	x13.4



Update motion



Update motion

$\downarrow$

$[\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}]$

$\downarrow$

Motion of the bodies ($\mathbf{r}_{cdg}, \dot{\mathbf{r}}_{cdg}, \omega$, etc.)

Remarks

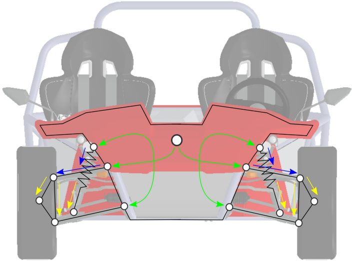
- Root-to-leafs procedure
- Data dependency between bodies
- Pipeline opportunity

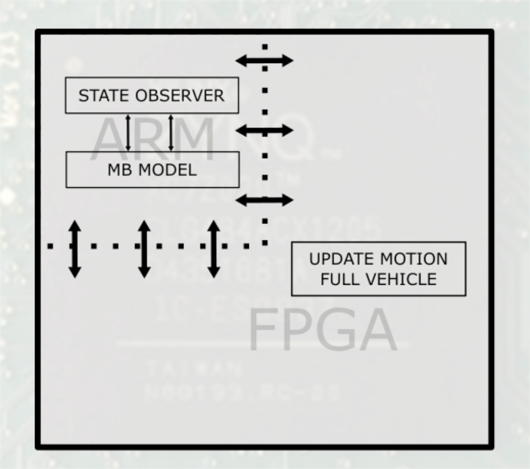
FPGA implementation: update motion (strategy 1)

Strategy 1

ARM	MB-based state observer
FPGA	Full update motion calc.

NOT ENOUGH FPGA RESOURCES



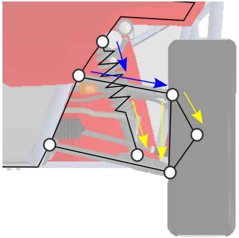


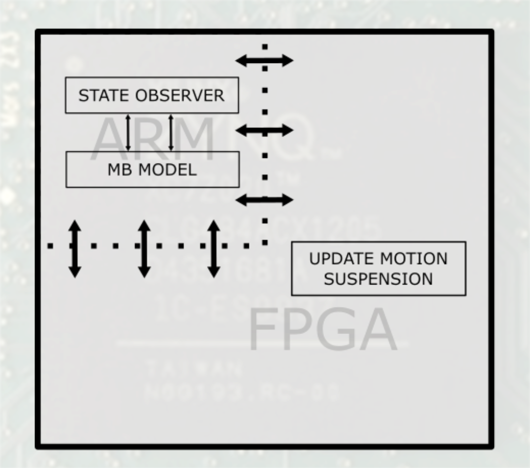
FPGA implementation: update motion (strategy 2)

Strategy 2

ARM — MB-based state observer

FPGA — Susp. update motion calc.





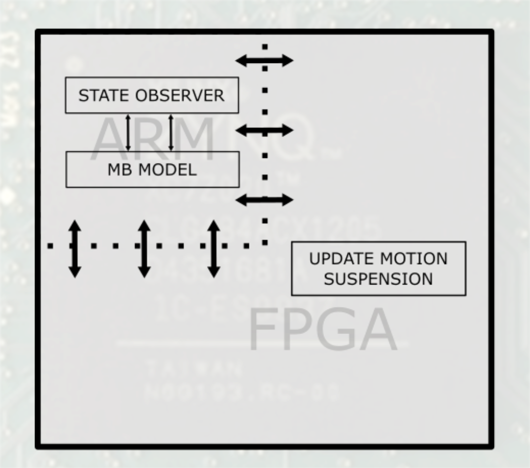
FPGA implementation: update motion (strategy 2)

Strategy 2

ARM — MB-based state observer

FPGA — Susp. update motion calc.

Implementation summary			
Parallel	Resources	Latency	Speed-up
No	Yes	3194	-
Pipeline	Yes	407	x7.9

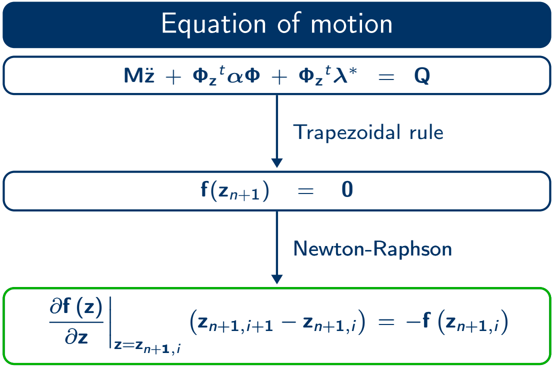


Solver of linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Solvers

- LU factorization
- QR factorization
- Gauss-Jordan
- Cholesky factorization



Equation of motion

$$\mathbf{M}\ddot{\mathbf{z}} + \boldsymbol{\Phi}_{\mathbf{z}}{}^{t}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{z}}{}^{t}\boldsymbol{\lambda}^{*} = \mathbf{Q}$$

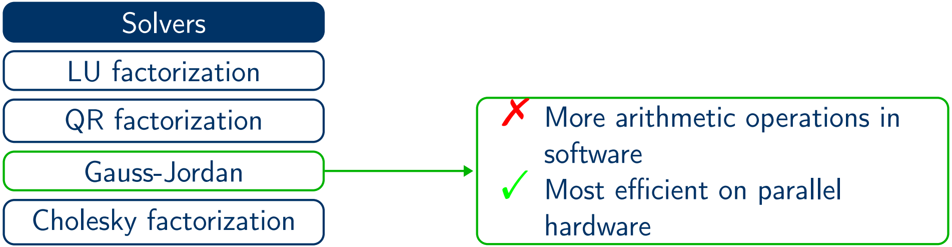
Trapezoidal rule

$$\mathbf{f}(\mathbf{z}_{n+1}) = \mathbf{0}$$

Newton-Raphson

$$\left.\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_{n+1,i}} \left(\mathbf{z}_{n+1,i+1} - \mathbf{z}_{n+1,i}\right) = -\mathbf{f}\left(\mathbf{z}_{n+1,i}\right)$$

Solver of linear system of equations



Solvers

- LU factorization
- QR factorization
- Gauss-Jordan
- Cholesky factorization

Gauss-Jordan →

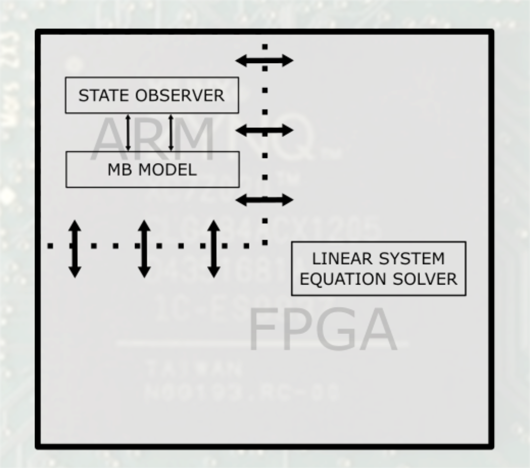
- ✗ More arithmetic operations in software
- ✓ Most efficient on parallel hardware

FPGA implementation: solver

Strategy 1

- ARM: MB-based state observer
- FPGA: Solver [5]

Implementation summary			
Parallel	Resources	Latency	Speed-up
No	Yes	928270	-
Option 1	No	10667	x87
Option 2	Yes	16428	x56.5



[5] J. P. David, "Low latency and division free Gauss–Jordan solver in floating point arithmetic," *Journal of Parallel and Distributed Computing*, vol. 106, pp. 185–193, Aug. 2017.

Outline

1 Introduction

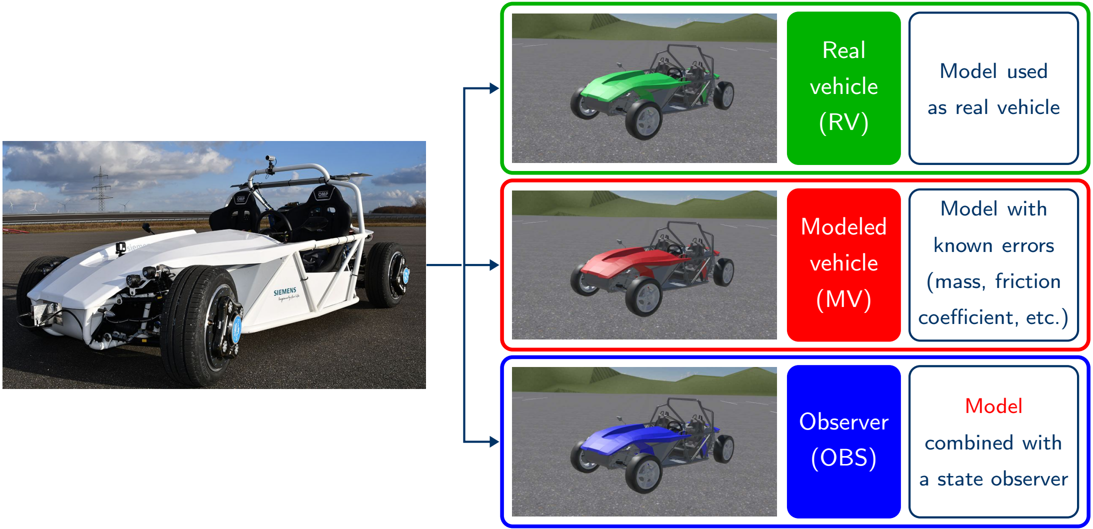
2 Model-based state observers

3 New generation embedded hardware

4 Use-case application

5 Conclusions and future work

Methodology



Real vehicle (RV)

Model used as real vehicle

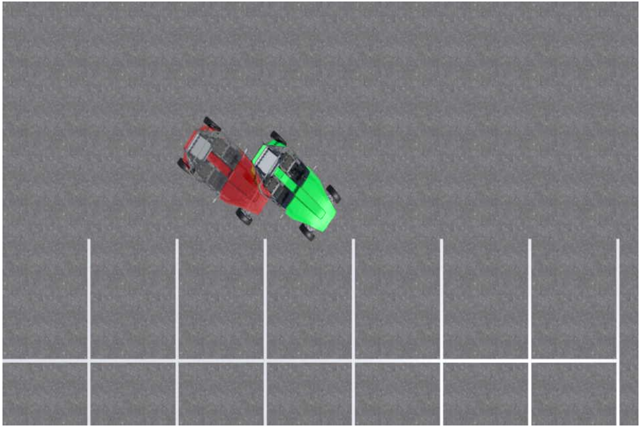
Modeled vehicle (MV)

Model with known errors (mass, friction coefficient, etc.)

Observer (OBS)

Model combined with a state observer

Complete vehicle MB model: maneuver

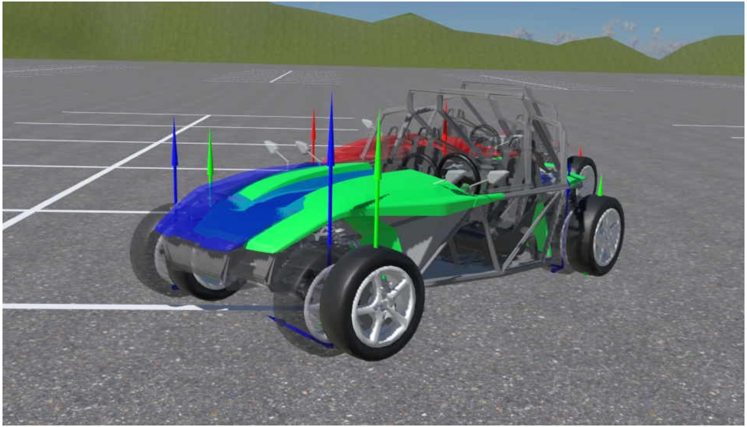


Maneuver

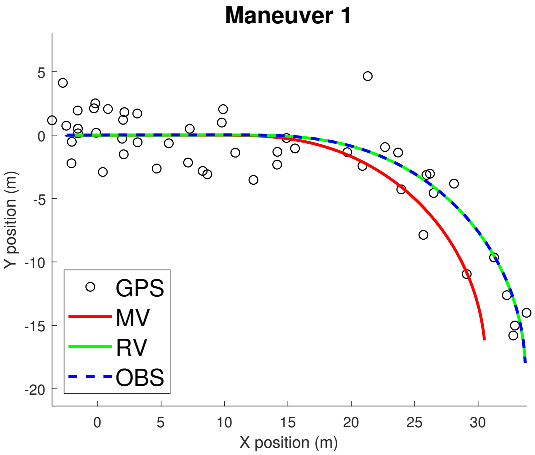
- Constant throttle position
- Constant steer at 5 s

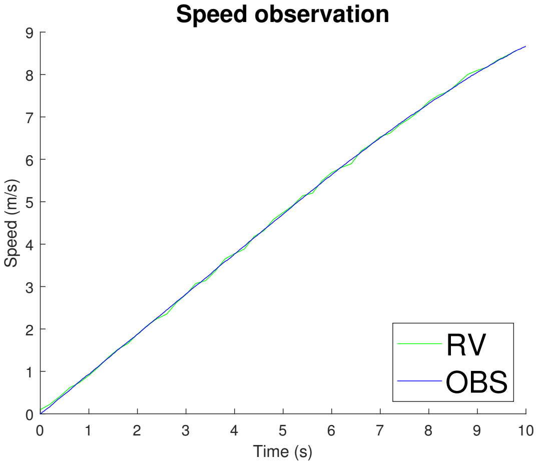
Real vehicle	Model
$Mass = 600kg$ $\mu = 0.8$	$Mass = 700kg$ $\mu = 1.0$

Estimations (errorEKF + complete vehicle MB model)

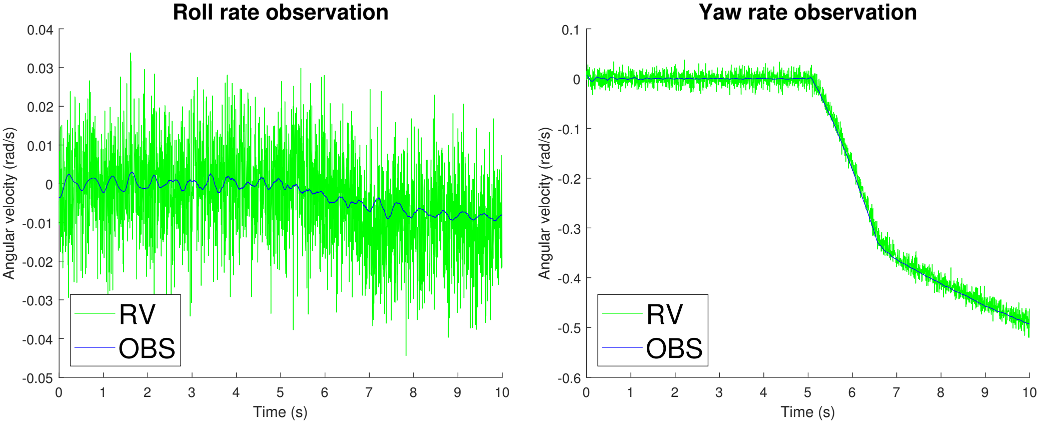


Estimations (errorEKF + complete vehicle MB model): sensors

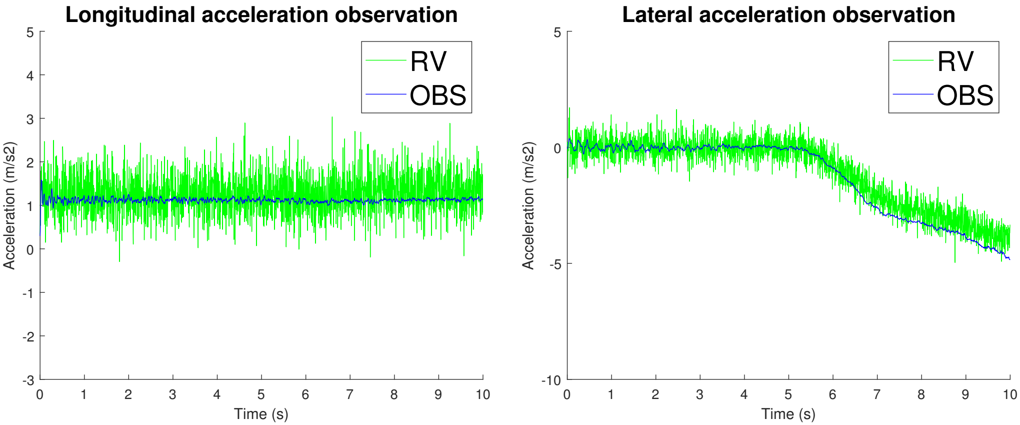




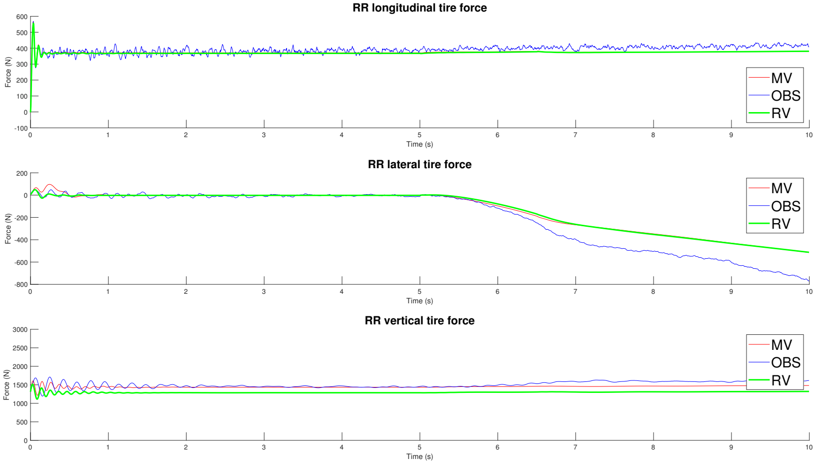
Estimations (errorEKF + complete vehicle MB model): sensors



Estimations (errorEKF + complete vehicle MB model): sensors



Estimations (errorEKF + complete vehicle MB model): tire forces



Implementation (errorEKF + complete vehicle MB model)

Summary Report

Version		Simulation Time (s)	Time Step (s)	Elapsed Time (s)	Average of Iterations	Tolerance
ARM	FPGA					
Full OBS	-	10	0.004	81.870	9.083	10^{-5}
Full OBS	-	10	0.004	38.284	1.419	10^{-3}
OBS	GJ	10	0.004	64.957	8.652	10^{-5}
OBS	GJ	10	0.004	33.459	1.388	10^{-3}
OBS	Inidv. Mass Matrix	10	0.004	83.622	9.153	10^{-5}
OBS	Susp. Post-Process	10	0.004	128.009	16.792	10^{-5}

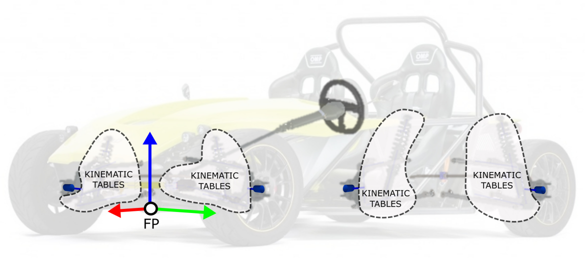
Implementation (errorEKF + complete vehicle MB model)

Summary Report

Version		Simulation Time (s)	Time Step (s)	Elapsed Time (s)	Average of Iterations	Tolerance
ARM	FPGA					
Full OBS	-	10	0.004	81.870	9.083	10^{-5}
Full OBS	-	10	0.004	38.284	1.419	10^{-3}
OBS	GJ	10	0.004	64.957	8.652	10^{-5}
OBS	GJ	10	0.004	33.459	1.388	10^{-3}
OBS	Inidv. Mass Matrix	10	0.004	83.622	9.153	10^{-5}
OBS	Susp. Post-Process	10	0.004	128.009	16.792	10^{-5}

NO REAL-TIME PERFORMANCE

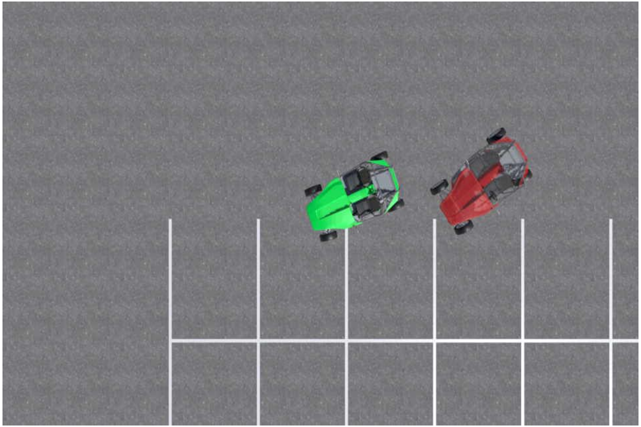
Simplified vehicle MB model



Summary	
DOFs	14
Suspension	Kinematic tables [6]
Rel. Coords.	14
Bodies	9
Tire model	TMeasy

[6] J. Cuadrado, D. Vilela, I. Iglesias, A. Martín, and A. Peña, "A multibody model to assess the effect of automotive motor in-wheel configuration on vehicle stability and comfort," in *2013 ECCOMAS Thematic Conference on Multibody Dynamics*, p. 10, 2013.

Simplified vehicle MB model: maneuver 1

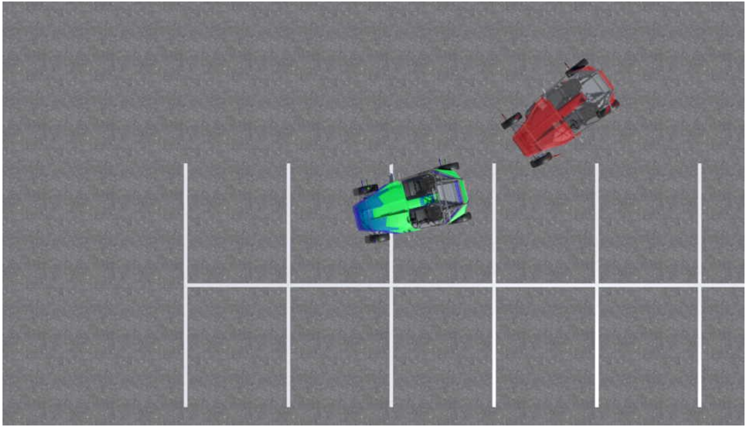


Maneuver

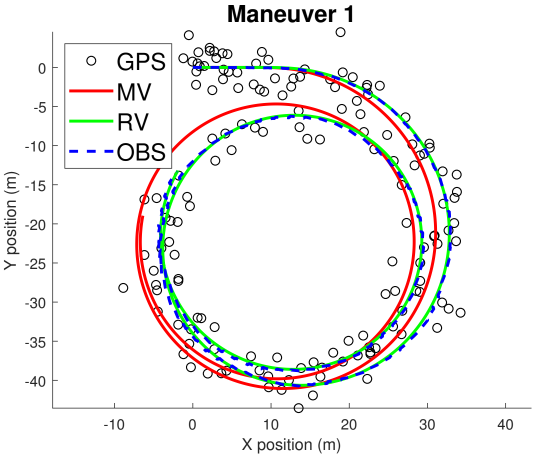
- Constant throttle position
- Constant steer at 5 s

Real vehicle	Model
$Mass = 600 kg$ $\mu = 0.8$	$Mass = 700 kg$ $\mu = 1.0$

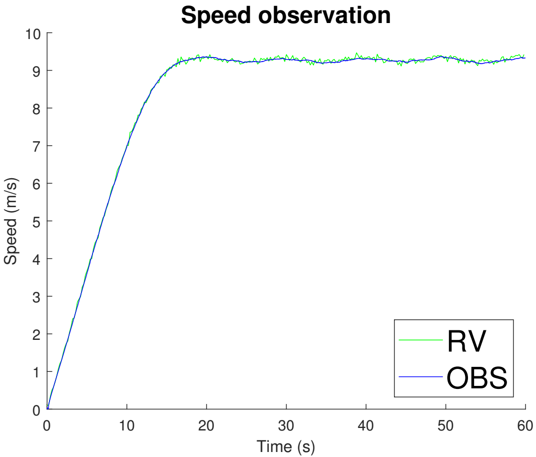
Estimations maneuver 1 (errorEKF + simplified vehicle MB model)



Estimations maneuver 1 (errorEKF + simplified vehicle MB model): sensors

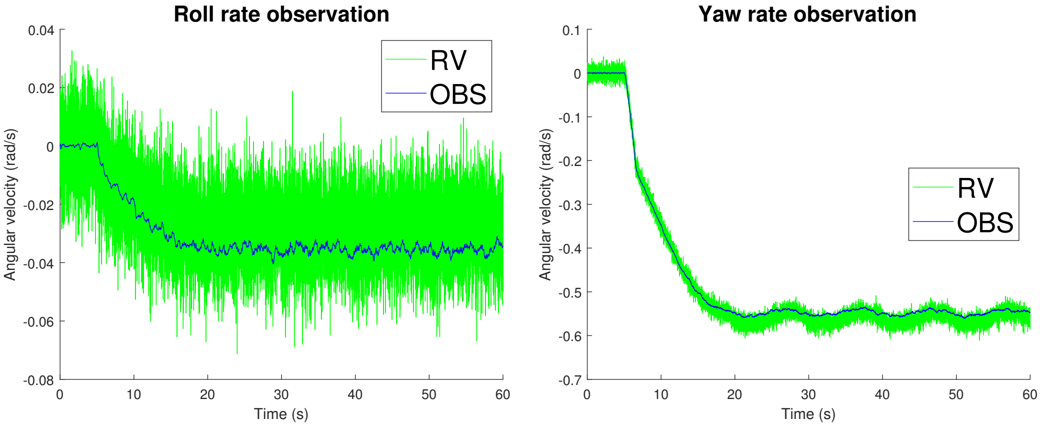


Maneuver 1

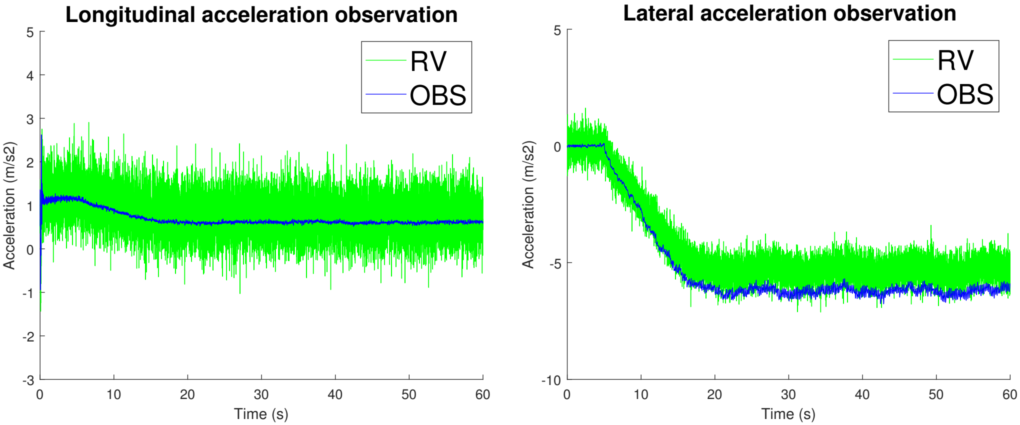


Speed observation

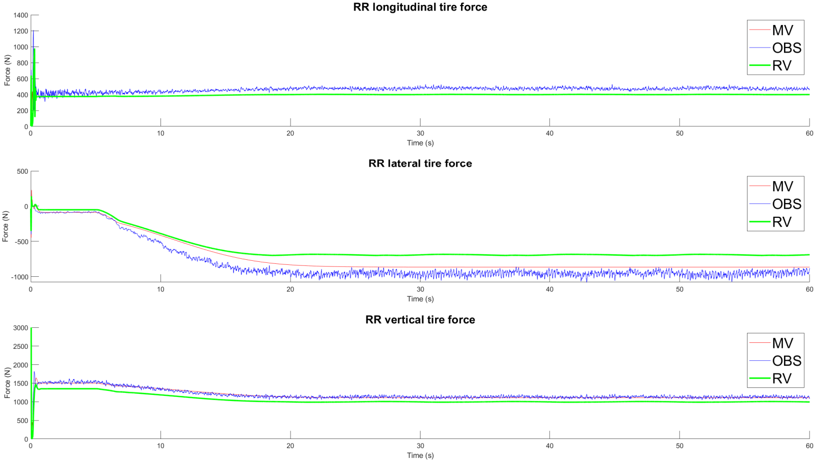
Estimations maneuver 1 (errorEKF + simplified vehicle MB model): sensors



Estimations maneuver 1 (errorEKF + simplified vehicle MB model): sensors



Estimations maneuver 1 (errorEKF + simplified vehicle MB model): tire forces



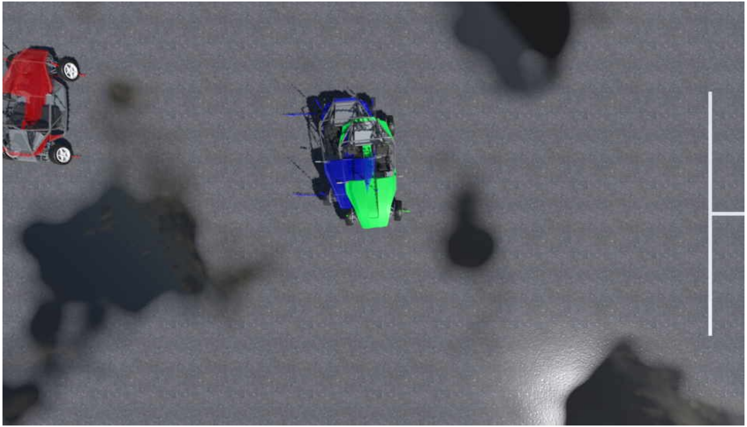
Simplified vehicle MB model: maneuver 2

Maneuver

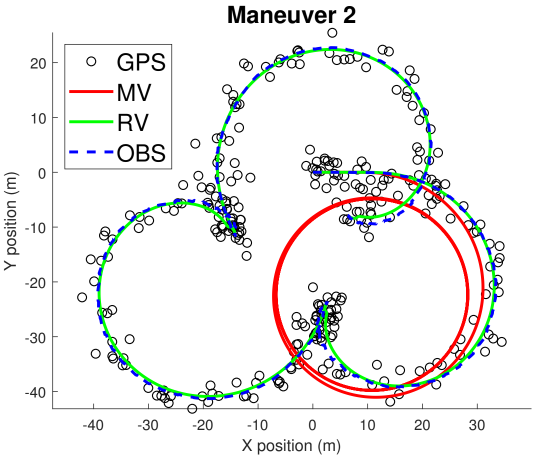
- Constant throttle position
- Constant steer at 5 s

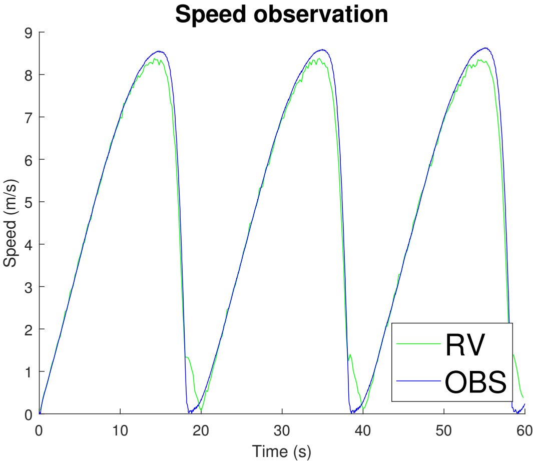
Real vehicle	Model
$Mass = 600kg$ $\mu = 0.6$	$Mass = 700kg$ $\mu = 1.0$

Estimations maneuver 2 (errorEKF + simplified vehicle MB model)

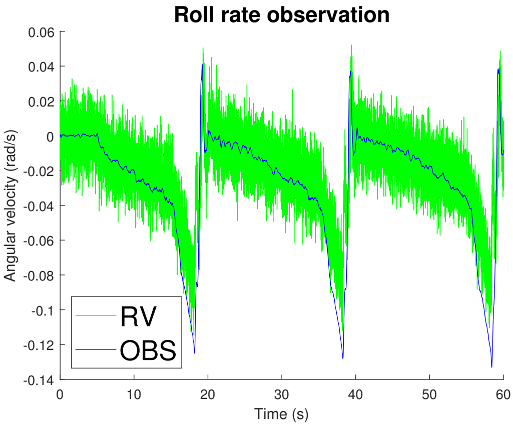


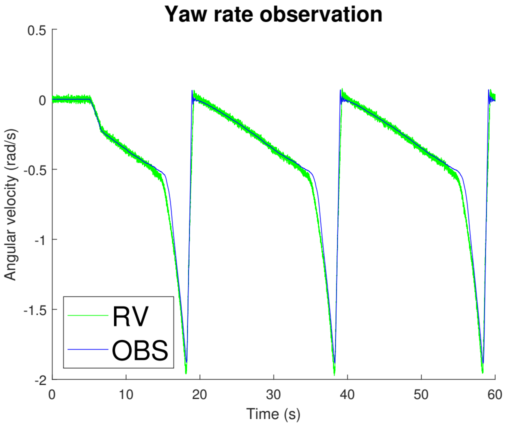
Estimations maneuver 2 (errorEKF + simplified vehicle MB model): sensors



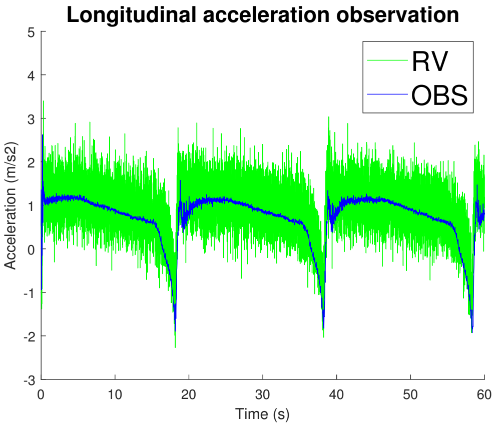


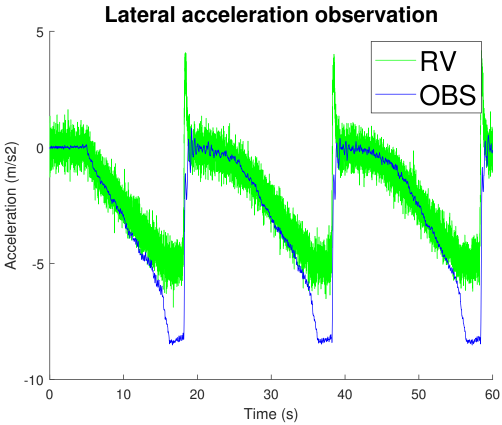
Estimations maneuver 2 (errorEKF + simplified vehicle MB model): sensors



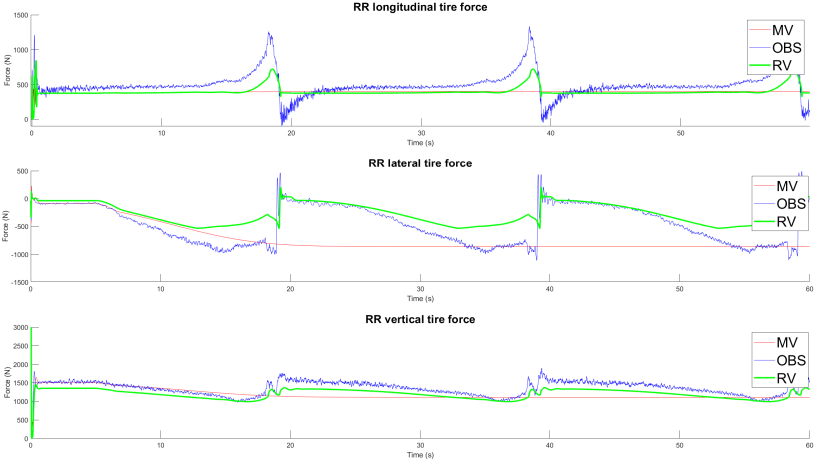


Estimations maneuver 2 (errorEKF + simplified vehicle MB model): sensors





Estimations maneuver 2 (errorEKF + simplified vehicle MB model): tire forces



Estimations (errorEKF + simplified vehicle MB model): RMSE

Root-mean-square error

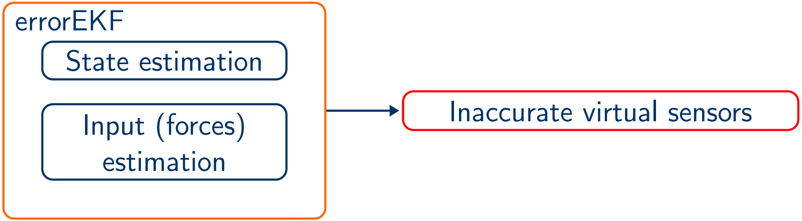
Magnitude	Maneuver 1 errorEKF	Maneuver 2 errorEKF	Sensor
Position (m)	0.1988	0.4023	1.9075
X accel. (m/s^2)	0.1319	0.3018	0.4492
Y accel. (m/s^2)	0.7923	1.5429	0.447
Z accel. (m/s^2)	0.3496	0.3662	0.4485
RR long. tire force (N)	76.09	165.32	-
RR lat. tire force (N)	237.69	265.58	-
RR vert. tire force (N)	144.25	180.09	-

errorEKF

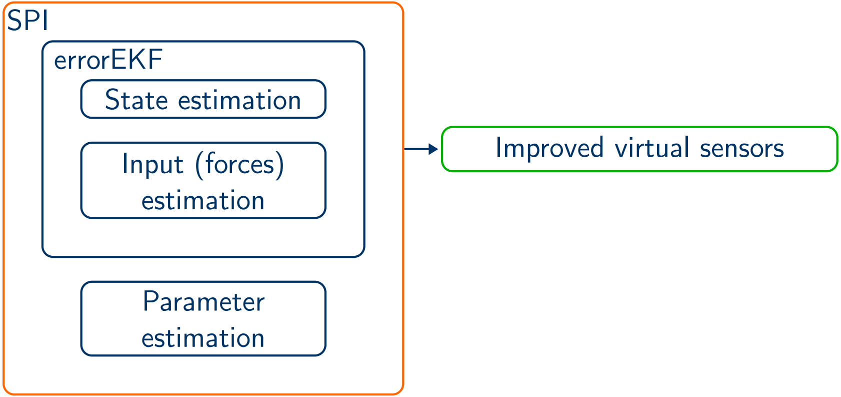
- ✓ Accurate position estimations
- ✓ Accurate longitudinal dynamics
- ✗ High error in lateral dynamics
- ✗ High error in tire forces

Mass and μ errors are not fully corrected

State-parameter-input observer



State-parameter-input observer



SPI

errorEKF

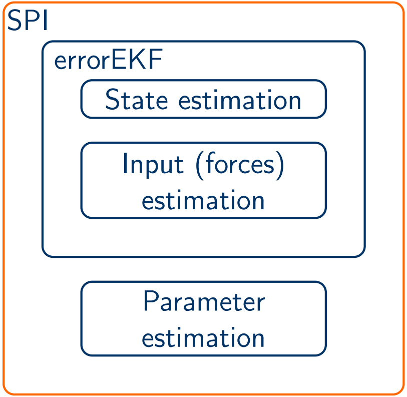
State estimation

Input (forces) estimation

Parameter estimation

Improved virtual sensors

State-parameter-input observer



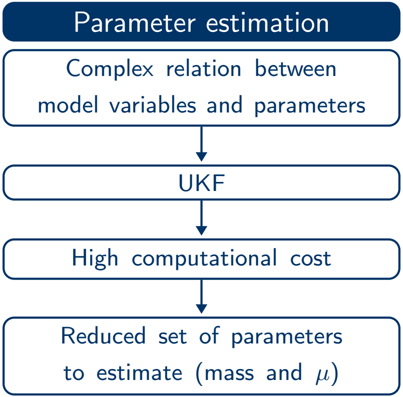
SPI

errorEKF

State estimation

Input (forces) estimation

Parameter estimation



Parameter estimation

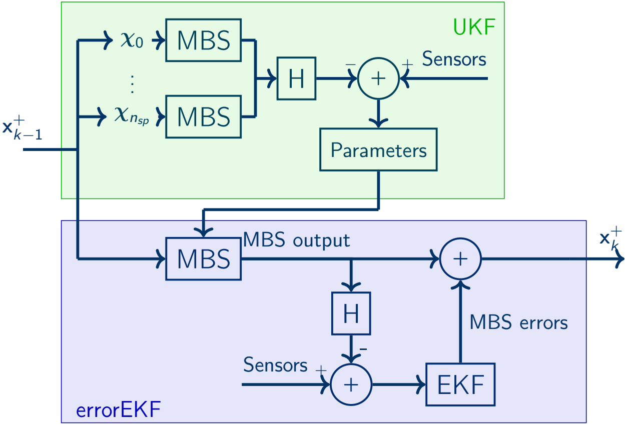
Complex relation between model variables and parameters

UKF

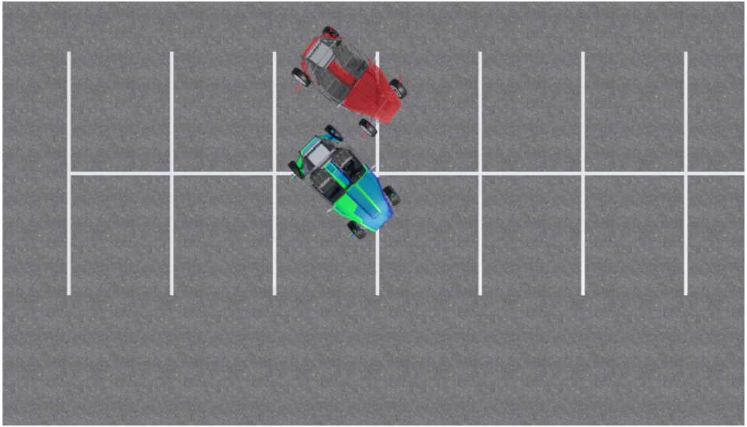
High computational cost

Reduced set of parameters to estimate (mass and μ)

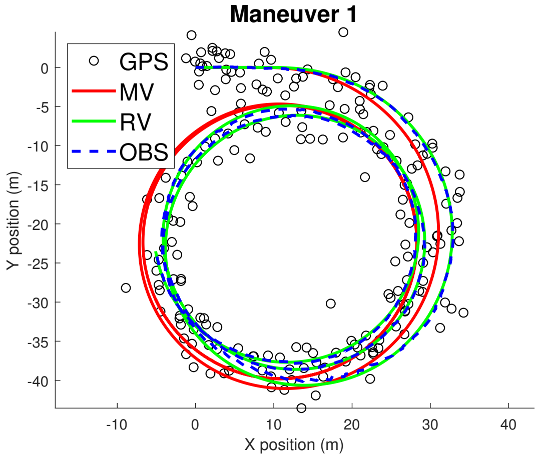
State-parameter-input observer

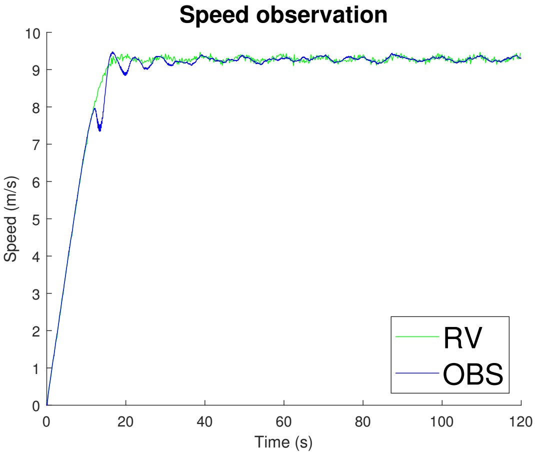


Estimations maneuver 1 (SPI + simplified vehicle MB model)

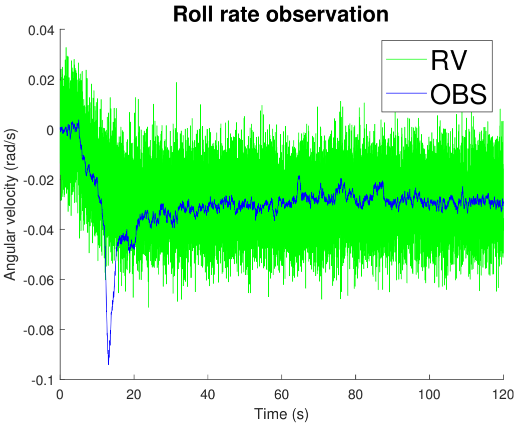


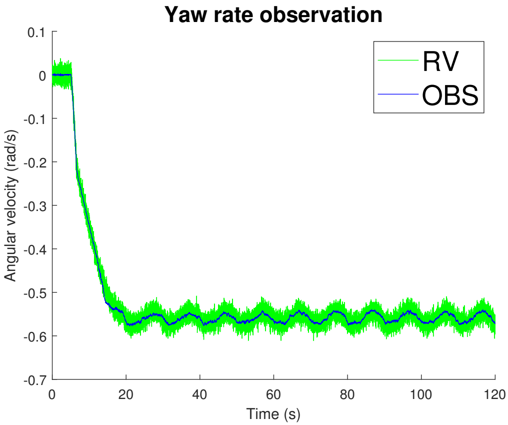
Estimations maneuver 1 (SPI + simplified vehicle MB model): sensors



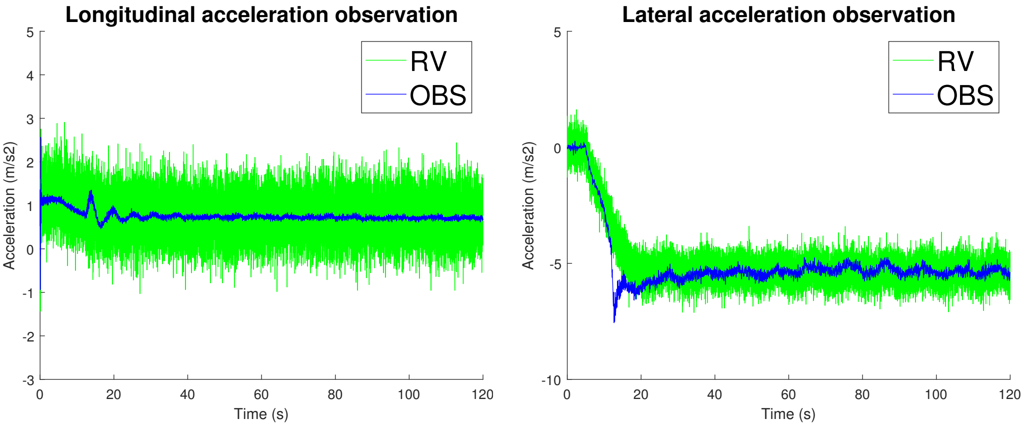


Estimations maneuver 1 (SPI + simplified vehicle MB model): sensors

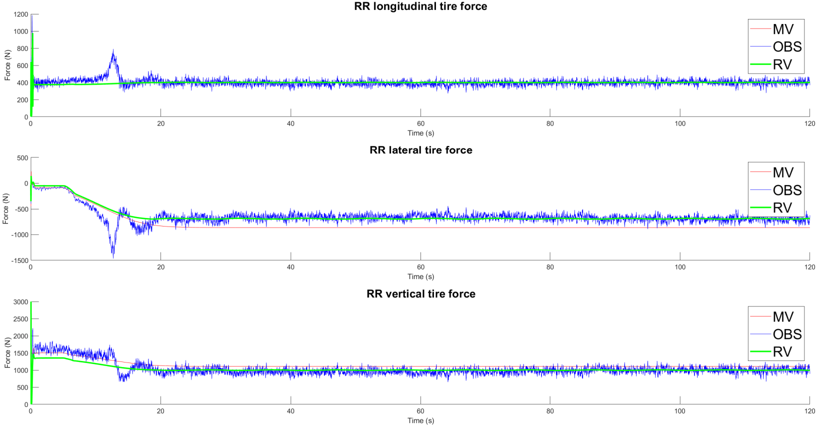




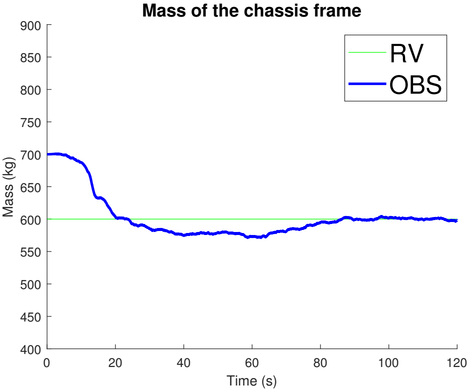
Estimations maneuver 1 (SPI + simplified vehicle MB model): sensors

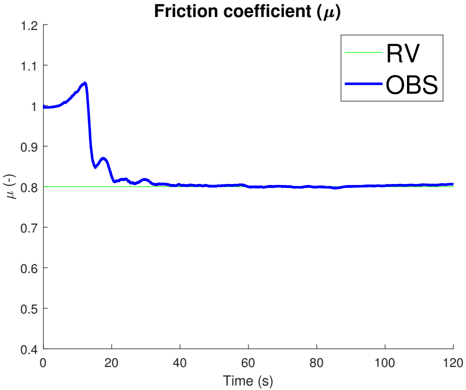


Estimations maneuver 1 (SPI + simplified vehicle MB model): tire forces



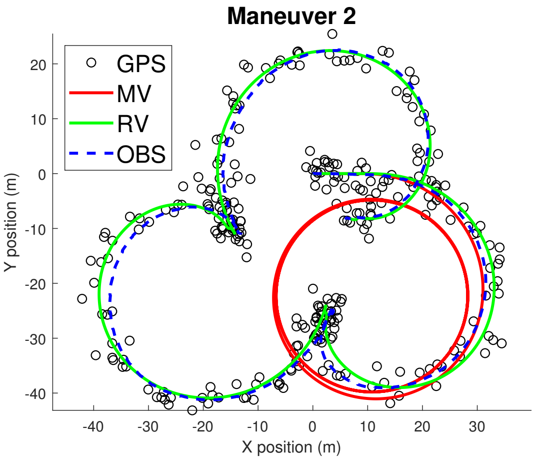
Estimations maneuver 1 (SPI + simplified vehicle MB model): parameters

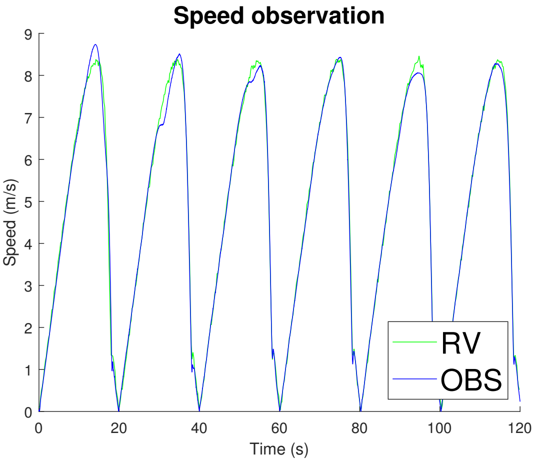




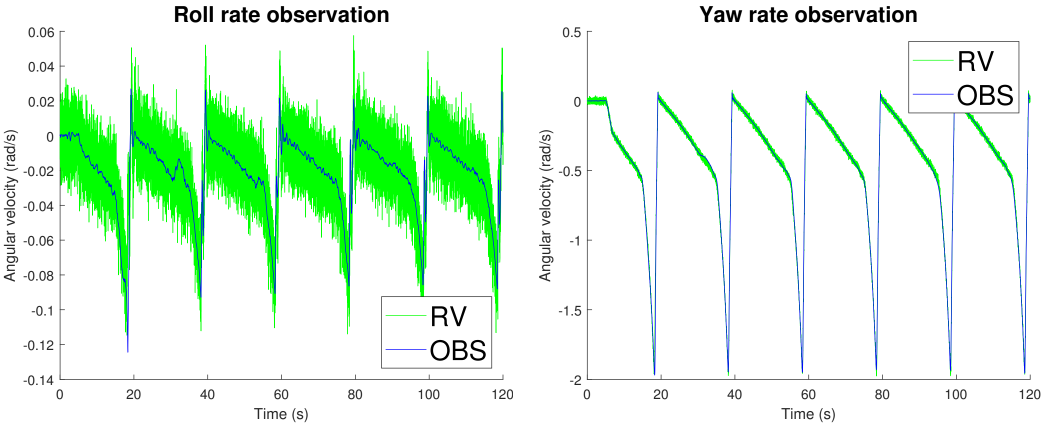
Estimations maneuver 2 (SPI + simplified vehicle MB model)

Estimations maneuver 2 (SPI + simplified vehicle MB model): sensors

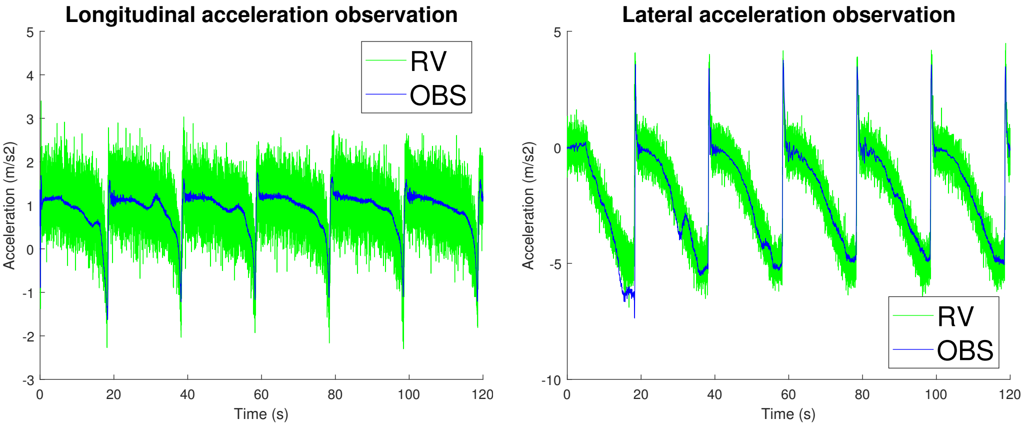




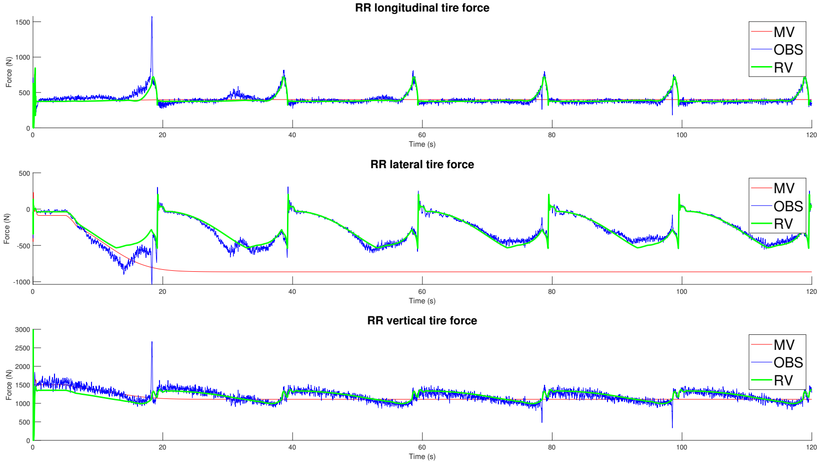
Estimations maneuver 2 (SPI + simplified vehicle MB model): sensors



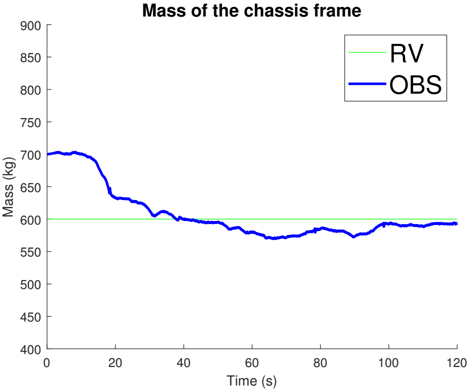
Estimations maneuver 2 (SPI + simplified vehicle MB model): sensors

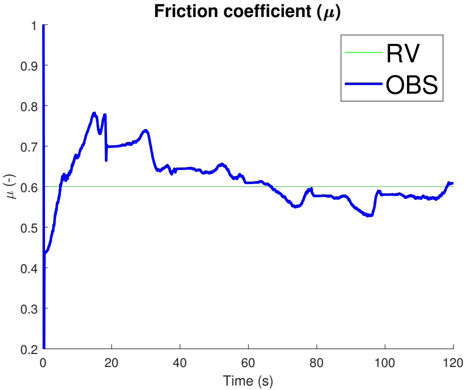


Estimations maneuver 2 (SPI + simplified vehicle MB model): tire forces



Estimations maneuver 2 (SPI + simplified vehicle MB model): parameters





Estimations (SPI + simplified vehicle MB model): RMSE

Root-mean-square error

Magnitude	Maneuver 1		Maneuver 2	
	SPI	errorEKF	SPI	errorEKF
Position (m)	0.2196	0.1988	0.4793	0.4023
X accel. (m/s^2)	0.0343	0.1319	0.0634	0.3018
Y accel. (m/s^2)	0.1551	0.7923	0.2750	1.5429
Z accel. (m/s^2)	0.1022	0.3496	0.1121	0.3662
RR long. tire force (N)	28.27	76.09	28.41	165.32
RR lat. tire force (N)	57.77	237.69	41.27	265.58
RR vert. tire force (N)	79.48	144.25	75.30	180.09

SPI

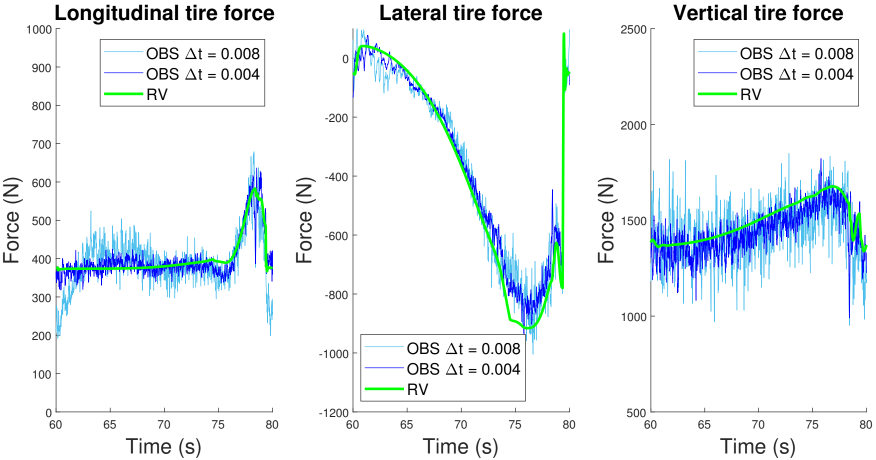
- ✓ Improved accuracy
- ✓ Long. tire force error from 41% to 7%
- ✓ Lat. tire force error from 50% to 9%
- ✓ Vert. tire force error from 18% to 8%
- ✗ Computational cost?

Implementation (errorEKF/SPI + simplified vehicle MB model)

Summary Report

Version		Simulation Time (s)	Time Step (s)	Elapsed Time (s)	Average of Iterations	Tolerance
ARM	FPGA					
Full OBS (errorEKF)	-	10	0.004	7.673	1.554	10^{-5}
Full OBS (SPI)	-	10	0.004	21.114	1.512	10^{-5}
OBS (errorEKF)	GJ	10	0.004	7.465	1.511	10^{-5}
OBS (SPI)	GJ	10	0.004	20.158	1.544	10^{-5}
Full OBS (SPI)	-	10	0.008	13.3456	3.055	10^{-5}
Full OBS (SPI)	-	10	0.008	9.381	1.046	$2 \cdot 10^{-4}$
OBS (SPI)	GJ	10	0.008	12.843	3.134	10^{-5}
OBS (SPI)	GJ	10	0.008	8.957	1.047	$2 \cdot 10^{-4}$

Implementation (errorEKF/SPI + simplified vehicle MB model)



Implementation (errorEKF/SPI + simplified vehicle MB model)

Root-mean-square error			
Magnitude	SPI ($\Delta t = 4\ ms$)	SPI ($\Delta t = 8\ ms$)	errorEKF ($\Delta t = 4\ ms$)
RR long. tire force (N)	28.41 (7%)	57.12 (14%)	165.32 (41%)
RR lat. tire force (N)	41.27 (9%)	63.06 (12%)	265.58 (50%)
RR vert. tire force (N)	75.30 (8%)	135.73 (13%)	180.09 (18%)

Outline

1 Introduction

2 Model-based state observers

3 New generation embedded hardware

4 Use-case application

5 Conclusions and future work

Conclusions

FPGA in multibody simulations

- FPGAs can be used for accelerating MB simulations
- FPGA guidelines:
  - Profile the code
  - Detect bottlenecks of the simulation
  - Analyze data dependencies
  - Check the available FPGA resources
  - Parallelize the code
- The amount of resources is a critical factor
- The model size and features affect to the final acceleration level

Virtual sensors based on MB models

- The errorEKF shows high efficiency, but low accuracy for tire-forces estimation
- The developed SPI observer increases the accuracy through parameter estimation
- Error reduction from:
  - 50% to 9% in lateral forces
  - 41% to 7% in longitudinal forces
  - 18% to 8% in vertical forces

Conclusions

Real-time performance on embedded hardware

- A full vehicle MB model is expensive for real-time applications on embedded hardware
- A simplified vehicle MB model offers higher efficiency
- Virtual sensors in real time are provided with the SPI observer combined with the simplified vehicle model:
  - Frequency of 125Hz
  - Around 10-15% error in tire force estimation

Virtual sensor framework

- MB modeling: MBScoder is improved with the addition of relative coordinates
- The state observer is compliant with the FMI 2.0 Standard:
  - High level of abstraction for new users
  - Easy integration with many tools

Future work

FPGA in multibody simulations

- Test devices with higher resources
- Develop a procedure for optimally select the best candidates of a MB simulation to be implemented on FPGAs

Virtual sensors based on MB models

- Replace the UKF of the SPI observer by an EKF to increase the computational efficiency
- Explore the tuning process of the filter noises
- Implement the observer in a real vehicle and test it on different maneuvers

Works derived from this thesis

- Published journal papers
  - A.J. Rodriguez, R. Pastorino, A. Carro-Lagoa, K. Janssens and M.A. Naya. Hardware acceleration of multibody simulations for real-time embedded applications. *Multibody System Dynamics* (2020).
- Submitted journal papers (under review)
  - A.J. Rodriguez, E. Sanjurjo, R. Pastorino and M.A. Naya. State, parameter and input observers based on multibody models and Kalman filters for vehicle dynamics. *Mechanical Systems and Signal Processing*.

Conference communications

- A.J. Rodriguez, R. Pastorino, M.A. Naya, E. Sanjurjo and W. Desmet. Real-time estimation based on multibody dynamics for automotive embedded heterogeneous computing. In *8th ECCOMAS Thematic Conference on Multibody Dynamics*, Prague, Czech Republic, June 2017.
- E. Sanjurjo, D. Dopico, M.A. Naya and A.J. Rodriguez. Indirect state and force estimator based on multibody models. In *8th ECCOMAS Thematic Conference on Multibody Dynamics*, Prague, Czech Republic, June 2017.
- A.J. Rodriguez, R. Pastorino, M.A. Naya and E. Sanjurjo. Virtual sensing on automotive embedded heterogeneous platforms. In *15th European Automotive Congress (EAEC 2017)*, Madrid, Spain, October 2017.
- A.J. Rodriguez, R. Pastorino, A. Luaces, E. Sanjurjo and M.A. Naya. Implementation of state observers based on multibody dynamics on automotive platforms in real-time. In *5th Joint Int. Conference on Multibody System Dynamics (IMSD 2018)*, Lisbon, Portugal, June 2018.
- E. Sanjurjo, A.J. Rodriguez, D. Dopico, A. Luaces and M.A. Naya. State and input observer for the multibody model of a car. In *5th Joint Int. Conference on Multibody System Dynamics (IMSD 2018)*, Lisbon, Portugal June, 2018.
- A.J. Rodriguez, R. Pastorino, E. Sanjurjo, A. Luaces and M.A. Naya. Implementación de Observador de Estados basado en Modelos Multicuerpo en Tiempo Real en Plataformas Embebidas. In *XXII Congreso Nacional de Ingeniería Mecánica*, Madrid, Spain, September 2018.

Implementation in embedded systems of state observers based on multibody dynamics

UNIVERSIDADE DA CORUÑA