Implementation in embedded systems of state observers based on multibody dynamics

Antonio J. Rodríguez

Doctoral thesis

Ferrol, June 25th, 2020
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
  - AEB
  - ACC
  - LC

Vehicle design
  - Model-based design
  - Vehicle testing

SENSORS
Use-Case: Wheel Force Transducers

Real sensors
- Instrumented rim
- Based on strain gauges
- Expensive

Virtual sensors
- Model-based
- Virtual environment
- Minimal set of sensors
Virtual Sensing

VEHICLE MODEL

DRIVER INPUTS

VIRTUAL SENSORS
Virtual Sensing

VEHICLE MODEL

DISTURBANCES
UNMODELED DYNAMICS
UNKNOWN PARAMETERS

DRIVER INPUTS
VIRTUAL SENSORS
DRIFT
Virtual Sensing

VEHICLE MODEL

STATE OBSERVER

DRIVER INPUTS

VIRTUAL SENSORS

SENSOR DATA
Virtual Sensing: Vehicle modeling

Vehicle Model

Analytical Models
- Low computational cost
- Reduced complexity
- Less vehicle parameters
- Lower accuracy
- Maneuver specific

Multibody Models
- High computational cost
- More virtual sensors
- More vehicle parameters
- Higher accuracy
- More versatile
Virtual Sensing: 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)

- In-vehicle embedded hardware
- Automotive standard compliant:
  - 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

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)

---

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

---

Semi-recursive method

Relative coord. $\rightarrow z$

$Z_i = Z_{i-1} + b_i \dot{z}_i$

Body coord. $\rightarrow Z = \begin{bmatrix} \dot{s} \\ \omega \end{bmatrix}$

Virtual power

$Z^T (\ddot{M} \dot{Z} - \ddot{Q}) = 0$

Equation of motion

$M \ddot{z} + \Phi_z^T \alpha \Phi + \Phi_z^T \lambda^* = \ddot{Q}$

$\text{ALI3P}$

Vehicle MB model

Summary

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOFs ($z^i$)</td>
<td>14</td>
</tr>
<tr>
<td>Steer</td>
<td>Kinematically guided</td>
</tr>
<tr>
<td>Rel. Coords. ($z^d$)</td>
<td>42</td>
</tr>
<tr>
<td>Bodies</td>
<td>29</td>
</tr>
<tr>
<td>Constraints</td>
<td>42</td>
</tr>
<tr>
<td>Tire model</td>
<td>TMeasy(^3)</td>
</tr>
</tbody>
</table>

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

---

**KF: errorEKF with force estimation**

1°: Prediction
\[
x = [\Delta z^i, \Delta \dot{z}^i, \Delta \ddot{z}^i]
\]
2°: Propagation
\[
\begin{align*}
\hat{x}^-_k &= 0 \\
P^-_k &= f_{x_{k-1}}P^+_k f_{x_{k-1}}^T + \Sigma^P
\end{align*}
\]
3°: KF Correction
\[
\begin{align*}
\hat{y}_k &= o_k - h(z_k, \dot{z}_k, \ddot{z}_k) \\
\Sigma_k &= h_x P^-_k h_x^T + \Sigma^S \\
K_k &= P^-_k h_x \Sigma^{-1}_k \\
\hat{x}^+_k &= \hat{x}^-_k + K_k \hat{y}_k \\
P^+_k &= (I - K_k h_x)P^-_k
\end{align*}
\]
4°: MB Correction
\[
\hat{x}^+_k \rightarrow [\Delta \hat{z}^d, \Delta \dot{\hat{z}}^d, \Delta \ddot{\hat{z}}^d]
\]
\[
\Delta \hat{Q}^d = 0
\]
\[
\Delta \hat{Q}^i = \hat{M}^i z^i - \hat{Q}^i
\]
\[
\Delta \hat{Q} = [\Delta \hat{Q}^i \Delta \hat{Q}^d]
\]
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
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**
- ✗ Lower computational power per core
- ✓ High computational power through parallelization
  - ✓ Improved area efficiency
  - ✓ Improved power efficiency
- ✗ Over-provisioned

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

---

Laboratorio de Ingeniería Mecánica
Universidade da Coruña

http://lim.ii.udc.es
Heterogeneous processors for scientific computing

**CORE A**
- Conventional embedded processor
  - Tailored for an application
  - Highest performance
  - Expensive
  - Long development times

**CORE B**
- ASIC
  - Programmable
  - Large number of cores
  - Best suited for larger systems
  - All cores are of the same type

- GPU
  - Programmable
  - Fit-for-purpose
  - Freedom for parallelization
  - Limited hardware resources

- FPGA
  - Programmable
  - Fit-for-purpose
  - Freedom for parallelization
  - Limited hardware resources
Field Programmable Gate Array (FPGA)

- 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
New generation embedded hardware  Modern hardware analysis

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

Application
Code profiling
Parallel intensive tasks
Non-parallel tasks

ARM
FPGA
**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

CLK

ITER

RD  CMP  WR
RD  CMP  WR
RD  CMP  WR
RD  CMP  WR
RD  CMP  WR
RD  CMP  WR
RD  CMP  WR
RD  CMP  WR

CLK

ITER $i$

RD  CMP  WR

ITER $i + 1$

RD  CMP  WR

CLK

ITER $i$

RD  CMP  WR

ITER $i + 1$

RD  CMP  WR

http://lim.ii.udc.es
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

Time Step

MB integration
- Predictor
- Update motion
- Mass matrix
- Forces
- Constraints
Solver

err < to

No

Yes

Projections

errorEKF
- Propagation
- Correction

MB correction
- Update motion
- Mass matrix
- Forces
- Force correction

Inputs

Outputs
Virtual sensors algorithm: profiling

Time Step

- MB integration
- Predictor
  - Update motion
  - Mass matrix
  - Forces
  - Constraints

Solver

- \( \text{err} < \text{tol} \)

No

Yes

Projections

- Error EKF
- Propagation
- Correction
  - MB correction
  - Mass matrix
  - Forces
  - Force correction

Outputs

- Inputs

Mass matrix: 11.06, 8.35
Update motion: 34.15, 33.35
Solve system: 12.89, 12.44

Laboratorio de Ingeniería Mecánica
Universidade da Coruña
http://lim.ii.udc.es
Mass matrix calculation

Body coordinates

Indiv. Mass Matrix

\[ \bar{M}_i = \begin{bmatrix} mI & -m\bar{g} \\ m\bar{g} & J - m\bar{g}\bar{g} \end{bmatrix} \]

Coordinates relation

\[ Z_i = Z_{i-1} + b_i\ddot{z}_i \]
\[ \dot{Z}_i = \dot{Z}_{i-1} + b_i\dddot{z}_i + d_i \]

Data dependency

Relative coordinates

\[ \begin{bmatrix} b_1 & 0 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & 0 & 0 & 0 \\ b_1 & 0 & 0 & b_4 & 0 & 0 \\ b_1 & 0 & 0 & b_4 & b_5 & 0 \\ b_1 & 0 & 0 & b_4 & 0 & b_6 \end{bmatrix} \]

Global. Mass Matrix

\[ \bar{M} = R^T \bar{M} R \]
FPGA implementation: mass matrix calculation (strategy 1)

**Strategy 1**

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

**NOT ENOUGH FPGA RESOURCES**

**Body coordinates**
- Indiv. Mass Matrix

**Coordinates relation**

**Relative coordinates**
- Global. Mass Matrix

**Laboratorio de Ingeniería Mecánica**
**Universidade da Coruña**

FPGA implementation: mass matrix calculation (strategy 2)

NOT ENOUGH FPGA RESOURCES

Body coordinates
Indiv. Mass Matrix
Coordinates relation
Relative coordinates
Global. Mass Matrix

Strategy 2
ARM MB-based state observer
FPGA Assembly mass matrix

STATE OBSERVER
MB MODEL
MASS MATRIX ASSEMBLY
FPGA implementation: mass matrix calculation (strategy 3)

Strategy 3
- ARM
- FPGA

MB-based state observer
- Indiv. mass matrices

Body coordinates
- Indiv. Mass Matrix

Coordinates relation
- Relative coordinates
  - Global. Mass Matrix

Implementation summary
- Parallel Resources
  - No
  - Yes
  - Latency
  - Speed-up

Unroll
- No
  - 24 x 29

Pipeline
- Yes
  - 52 x 13.4

http://lim.ii.udc.es
FPGA implementation: mass matrix calculation (strategy 3)

<table>
<thead>
<tr>
<th>Implementation summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Unroll</td>
</tr>
<tr>
<td>Pipeline</td>
</tr>
</tbody>
</table>

Strategy 3
- ARM
- FPGA
- MB-based state observer
- Indiv. mass matrices
New generation embedded hardware  FPGA implementations

Update motion

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

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

Remarks

- Root-to-leaves procedure
- Data dependency between bodies
- Pipeline opportunity
FPGA implementation: update motion (strategy 1)

Strategy 1
- ARM
- FPGA

MB-based state observer
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)

<table>
<thead>
<tr>
<th>Parallel</th>
<th>Resources</th>
<th>Latency</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>3194</td>
<td>-</td>
</tr>
<tr>
<td>Pipeline</td>
<td>Yes</td>
<td>407</td>
<td>x7.9</td>
</tr>
</tbody>
</table>
**Solver of linear system of equations**

\[ \mathbf{A} \mathbf{x} = \mathbf{b} \]

### Solvers
- LU factorization
- QR factorization
- Gauss-Jordan
- Cholesky factorization

### Equation of motion
\[
\mathbf{M}\ddot{\mathbf{z}} + \mathbf{\Phi}_z^t \alpha \mathbf{\Phi} + \mathbf{\Phi}_z^t \lambda^* = \mathbf{Q}
\]

- Trapezoidal rule

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

- Newton-Raphson

\[
\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \bigg|_{\mathbf{z} = \mathbf{z}_{n+1}, i} (\mathbf{z}_{n+1,i+1} - \mathbf{z}_{n+1,i}) = -f(\mathbf{z}_{n+1,i})
\]
Solver of linear system of equations

Solvers

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

More arithmetic operations in software

Most efficient on parallel hardware
## FPGA implementation: solver

### Strategy 1
- **ARM**: MB-based state observer
- **FPGA**: Solver

---

### Implementation summary

<table>
<thead>
<tr>
<th>Parallel</th>
<th>Resources</th>
<th>Latency</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>928270</td>
<td>-</td>
</tr>
<tr>
<td>Option 1</td>
<td>No</td>
<td>10667</td>
<td>x87</td>
</tr>
<tr>
<td>Option 2</td>
<td>Yes</td>
<td>16428</td>
<td>x56.5</td>
</tr>
</tbody>
</table>

---

Outline

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

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

Methodology

Laboratorio de Ingeniería Mecánica
Universidade da Coruña

http://lim.ii.udc.es
Complete vehicle MB model: maneuver

Maneuver
- Constant throttle position
- Constant steer at 5 s

Real vehicle
- Mass = 600 kg
- \( \mu = 0.8 \)

Model
- Mass = 700 kg
- \( \mu = 1.0 \)
Estimations (errorEKF + complete vehicle MB model)
Estimations (errorEKF + complete vehicle MB model): sensors

Maneuver 1

Speed observation
Estimations (errorEKF + complete vehicle MB model): sensors

Roll rate observation

Yaw rate observation
Estimations (errorEKF + complete vehicle MB model): sensors

Longitudinal acceleration observation

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

- RR longitudinal tire force
- RR lateral tire force
- RR vertical tire force
### Implementation (errorEKF + complete vehicle MB model)

<table>
<thead>
<tr>
<th>Version</th>
<th>ARM</th>
<th>FPGA</th>
<th>Simulation Time (s)</th>
<th>Time Step (s)</th>
<th>Elapsed Time (s)</th>
<th>Average of Iterations</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full OBS</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>0.004</td>
<td>81.870</td>
<td>9.083</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Full OBS</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>0.004</td>
<td>38.284</td>
<td>1.419</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>OBS</td>
<td>GJ</td>
<td>-</td>
<td>10</td>
<td>0.004</td>
<td>64.957</td>
<td>8.652</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>OBS</td>
<td>GJ</td>
<td>-</td>
<td>10</td>
<td>0.004</td>
<td>33.459</td>
<td>1.388</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>OBS Inidv. Mass Matrix</td>
<td></td>
<td>-</td>
<td>10</td>
<td>0.004</td>
<td>83.622</td>
<td>9.153</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>OBS Susp. Post-Process</td>
<td></td>
<td>-</td>
<td>10</td>
<td>0.004</td>
<td>128.009</td>
<td>16.792</td>
<td>$10^{-5}$</td>
</tr>
</tbody>
</table>
## Implementation (errorEKF + complete vehicle MB model)

<table>
<thead>
<tr>
<th>Version</th>
<th>Simulation Time (s)</th>
<th>Time Step (s)</th>
<th>Elapsed Time (s)</th>
<th>Average of Iterations</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM - FPGA</td>
<td>10</td>
<td>0.004</td>
<td>81.870</td>
<td>9.083</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Full OBS - GJ</td>
<td>10</td>
<td>0.004</td>
<td>38.284</td>
<td>1.419</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>OBS - GJ</td>
<td>10</td>
<td>0.004</td>
<td>64.957</td>
<td>8.652</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>OBS - Invid. Mass Matrix</td>
<td>10</td>
<td>0.004</td>
<td>33.459</td>
<td>1.388</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>OBS - Susp. Post-Process</td>
<td>10</td>
<td>0.004</td>
<td>83.622</td>
<td>9.153</td>
<td>$10^{-5}$</td>
</tr>
</tbody>
</table>

**NO REAL-TIME PERFORMANCE**
Use-case application  Simplified vehicle model

Simplified vehicle MB model

Summary

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DOFs</td>
<td>14</td>
</tr>
<tr>
<td>Suspension</td>
<td></td>
</tr>
<tr>
<td>Rel. Coords.</td>
<td>14</td>
</tr>
<tr>
<td>Bodies</td>
<td>9</td>
</tr>
<tr>
<td>Tire model</td>
<td>TMeasy</td>
</tr>
</tbody>
</table>

---

Simplified vehicle MB model: maneuver 1

**Maneuver**
- Constant throttle position
- Constant steer at 5 s

**Real vehicle**
- Mass = 600 kg
- $\mu = 0.8$

**Model**
- Mass = 700 kg
- $\mu = 1.0$
Estimations maneuver 1 (errorEKF + simplified vehicle MB model)
Estimations maneuver 1 (errorEKF + simplified vehicle MB model): sensors
Estimations maneuver 1 (errorEKF + simplified vehicle MB model): sensors

Roll rate observation

![Roll rate observation graph]

Yaw rate observation

![Yaw rate observation graph]
Estimations maneuver 1 (errorEKF + simplified vehicle MB model): sensors

Longitudinal acceleration observation

Lateral acceleration observation
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**
- Mass = 600 kg
- $\mu = 0.6$

**Model**
- Mass = 700 kg
- $\mu = 1.0$
Estimations maneuver 2 (errorEKF + simplified vehicle MB model)
Estimations maneuver 2 (errorEKF + simplified vehicle MB model): sensors

Maneuver 2

GPS
MV
RV
OBS

Speed observation

RV
OBS
Use-case application  Simplified vehicle model

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

Roll rate observation

Yaw rate observation

Laboratorio de Ingeniería Mecánica
Universidade da Coruña
Estimations maneuver 2 (errorEKF + simplified vehicle MB model): sensors

**Longitudinal acceleration observation**

- RV
- OBS

**Lateral acceleration observation**

- RV
- OBS

Laboratorio de Ingeniería Mecánica
Universidade da Coruña

http://lim.ii.udc.es
Estimations maneuver 2 (errorEKF + simplified vehicle MB model): tire forces
Estimations (errorEKF + simplified vehicle MB model): RMSE

<table>
<thead>
<tr>
<th>Magnitude</th>
<th>Maneuver 1 errorEKF</th>
<th>Maneuver 2 errorEKF</th>
<th>Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position (m)</td>
<td>0.1988</td>
<td>0.4023</td>
<td>1.9075</td>
</tr>
<tr>
<td>X accel. (m/s²)</td>
<td>0.1319</td>
<td>0.3018</td>
<td>0.4492</td>
</tr>
<tr>
<td>Y accel. (m/s²)</td>
<td>0.7923</td>
<td>1.5429</td>
<td>0.447</td>
</tr>
<tr>
<td>Z accel. (m/s²)</td>
<td>0.3496</td>
<td>0.3662</td>
<td>0.4485</td>
</tr>
<tr>
<td>RR long. tire force (N)</td>
<td>76.09</td>
<td>165.32</td>
<td>-</td>
</tr>
<tr>
<td>RR lat. tire force (N)</td>
<td>237.69</td>
<td>265.58</td>
<td>-</td>
</tr>
<tr>
<td>RR vert. tire force (N)</td>
<td>144.25</td>
<td>180.09</td>
<td>-</td>
</tr>
</tbody>
</table>

- **errorEKF**
  - Accurate position estimations
  - Accurate longitudinal dynamics
  - High error in lateral dynamics
  - High error in tire forces

Mass and $\mu$ errors are not fully corrected
State-parameter-input observer

- **Error EKF**
  - State estimation
  - Input (forces) estimation

- Inaccurate virtual sensors
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 \( \mu \))
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

Roll rate observation

Yaw rate observation
Estimations maneuver 1 (SPI + simplified vehicle MB model): sensors
Estimations maneuver 1 (SPI + simplified vehicle MB model): tire forces
Estimaciones maniobra 1 (SPI + modelo simplificado de vehículo MB): parámetros

Mass of the chassis frame

Friction coefficient ($\mu$)

Laboratorio de Ingeniería Mecánica
Universidade da Coruña

http://lim.ii.udc.es
Estimations maneuver 2 (SPI + simplified vehicle MB model)
Estimations maneuver 2 (SPI + simplified vehicle MB model): sensors

Maneuver 2

Speed observation

Laboratorio de Ingeniería Mecánica
Universidade da Coruña

http://lim.ii.udc.es
Estimations maneuver 2 (SPI + simplified vehicle MB model): sensors
Estimations maneuver 2 (SPI + simplified vehicle MB model): sensors

Longitudinal acceleration observation

Lateral acceleration observation

RV
OBS

Laboratorio de Ingeniería Mecánica
Universidade da Coruña

http://lim.ii.udc.es
Estimations maneuver 2 (SPI + simplified vehicle MB model): tire forces

- RR longitudinal tire force
- RR lateral tire force
- RR vertical tire force
Estimations maneuver 2 (SPI + simplified vehicle MB model): parameters

Mass of the chassis frame

Friction coefficient ($\mu$)
## Estimations (SPI + simplified vehicle MB model): RMSE

### Root-mean-square error

<table>
<thead>
<tr>
<th>Magnitude</th>
<th>Maneuver 1</th>
<th>Maneuver 2</th>
<th>SPI error</th>
<th>SPI error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position (m)</td>
<td>0.2196</td>
<td>0.4793</td>
<td>0.1988</td>
<td>0.4023</td>
</tr>
<tr>
<td>X accel. (m/s²)</td>
<td>0.0343</td>
<td>0.0634</td>
<td>0.1319</td>
<td>0.3018</td>
</tr>
<tr>
<td>Y accel. (m/s²)</td>
<td>0.1551</td>
<td>0.2750</td>
<td>0.7923</td>
<td>1.5429</td>
</tr>
<tr>
<td>Z accel. (m/s²)</td>
<td>0.1022</td>
<td>0.1121</td>
<td>0.3496</td>
<td>0.3662</td>
</tr>
<tr>
<td>RR long. tire force (N)</td>
<td>28.27</td>
<td>0.1121</td>
<td>76.09</td>
<td>0.3662</td>
</tr>
<tr>
<td>RR lat. tire force (N)</td>
<td>57.77</td>
<td>28.41</td>
<td>237.69</td>
<td>165.32</td>
</tr>
<tr>
<td>RR vert. tire force (N)</td>
<td>79.48</td>
<td>75.30</td>
<td>144.25</td>
<td>180.09</td>
</tr>
</tbody>
</table>

### 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)

<table>
<thead>
<tr>
<th>Version</th>
<th>ARM</th>
<th>FPGA</th>
<th>Simulation Time (s)</th>
<th>Time Step (s)</th>
<th>Elapsed Time (s)</th>
<th>Average of Iterations</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full OBS (errorEKF)</td>
<td>-</td>
<td></td>
<td>10</td>
<td>0.004</td>
<td>7.673</td>
<td>1.554</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Full OBS (SPI)</td>
<td>-</td>
<td></td>
<td>10</td>
<td>0.004</td>
<td>21.114</td>
<td>1.512</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>OBS (errorEKF) GJ</td>
<td>GJ</td>
<td></td>
<td>10</td>
<td>0.004</td>
<td>7.465</td>
<td>1.511</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>OBS (SPI) GJ</td>
<td>GJ</td>
<td></td>
<td>10</td>
<td>0.004</td>
<td>20.158</td>
<td>1.544</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Full OBS (SPI)</td>
<td>-</td>
<td></td>
<td>10</td>
<td>0.008</td>
<td>13.3456</td>
<td>3.055</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Full OBS (SPI)</td>
<td>-</td>
<td></td>
<td>10</td>
<td>0.008</td>
<td>9.381</td>
<td>1.046</td>
<td>$2 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>OBS (SPI) GJ</td>
<td>GJ</td>
<td></td>
<td>10</td>
<td>0.008</td>
<td>12.843</td>
<td>3.134</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>OBS (SPI) GJ</td>
<td>GJ</td>
<td></td>
<td>10</td>
<td>0.008</td>
<td>8.957</td>
<td>1.047</td>
<td>$2 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>
Implementation (errorEKF/SPI + simplified vehicle MB model)

Longitudinal tire force

Lateral tire force

Vertical tire force

OBS $\Delta t = 0.008$

OBS $\Delta t = 0.004$

RV
### Implementation (errorEKF/SPI + simplified vehicle MB model)

<table>
<thead>
<tr>
<th>Magnitude</th>
<th>SPI ($\Delta t = 4$ ms)</th>
<th>SPI ($\Delta t = 8$ ms)</th>
<th>errorEKF ($\Delta t = 4$ ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR long. tire force (N)</td>
<td>28.41 (7%)</td>
<td>57.12 (14%)</td>
<td>165.32 (41%)</td>
</tr>
<tr>
<td>RR lat. tire force (N)</td>
<td>41.27 (9%)</td>
<td>63.06 (12%)</td>
<td>265.58 (50%)</td>
</tr>
<tr>
<td>RR vert. tire force (N)</td>
<td>75.30 (8%)</td>
<td>135.73 (13%)</td>
<td>180.09 (18%)</td>
</tr>
</tbody>
</table>
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
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 and future work

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
Conclusions and future work

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**

- **Submitted journal papers (under review)**
Conclusions and future work

Conference communications


Laboratorio de Ingeniería Mecánica
Universidade da Coruña
http://lim.ii.udc.es
Implementation in embedded systems of state observers based on multibody dynamics