An FPGA based sensor fusion algorithm for IMU data processing

Pietro Pennestrì

UNIVERSITY OF TWENTE.

Agenda

- Project Background and Introduction
- Sensor Fusion
  - Introduction to IMU sensor Fusion
  - Proposed sensor fusion algorithm
  - Proposed FPGA architecture
- Approximation of `ISQRT` and `SQRT`
  - Polynomial approximation
  - Proposed FPGA architecture
- Conclusion and Future Work

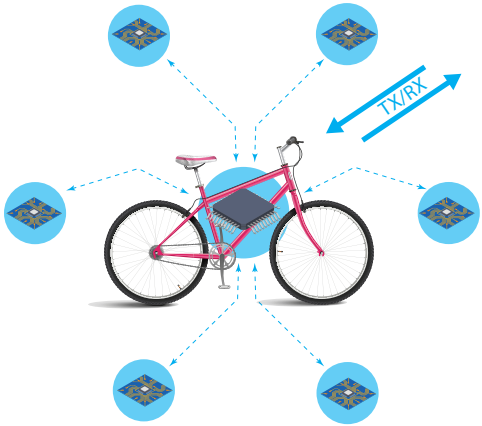
UNIVERSITY OF TWENTE.

Project Background

- The bike is recognized as a sustainable and energy-efficient method of transport.
- Although mainly limited to electric motor power control and to aftermarket accessories, there is a generalized trend toward bikes digitalization.
- The growing trend is witnessed by the wide use in commercial advertisement of the term **Smart Bike**.

UNIVERSITY OF TWENTE.

Smart Bike Proposed Definition



Smart Bike

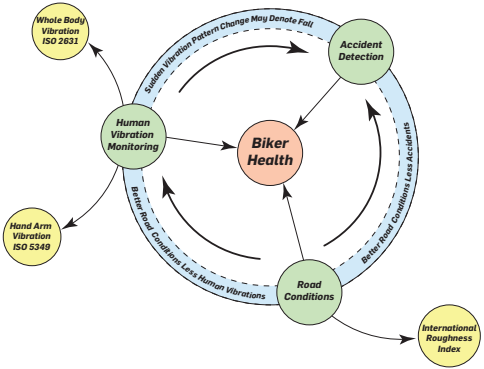
A **SMART BIKE** is a bike equipped with a variety of modular sensors & devices, connected to an onboard master unit, also capable of **data processing** and **transmission**, for the purpose of **enriching the user experience**, increasing his/her **awareness** and **safety**.

UNIVERSITY OF TWENTE.

V-AUDIT: a vibration monitoring device

V-AUDIT stands for **V**ibration **Audit** and is herein proposed as a low cost bike accessory composed of:

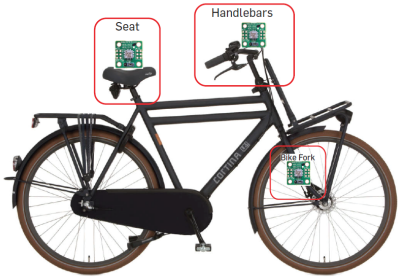
- a unit to collect and onboard process different Inertial Measurement Unit (IMU) sensors data
- different IMUs positioned at bike-human body interfaces (seat, handlebar).



UNIVERSITY OF TWENTE.

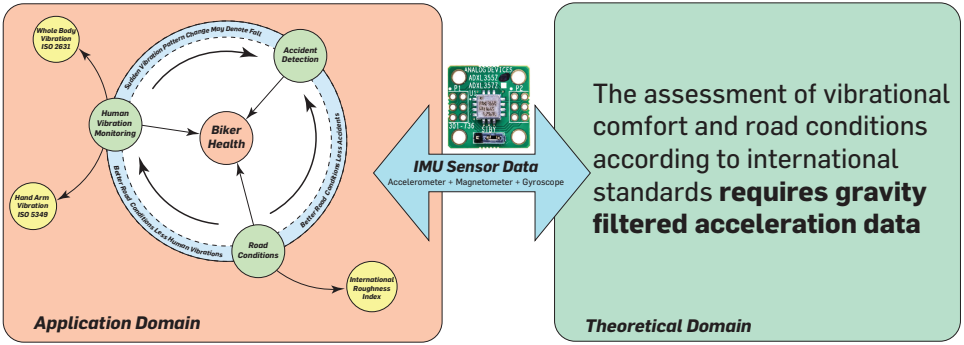
IMU

- The sensors on board 9-DoF IMU provide all data for the functional features of V-AUDIT. ✓
- Low Cost and easily installed on bikes. ✓
- Operations required: filtering and calibration. ✗



UNIVERSITY OF TWENTE.

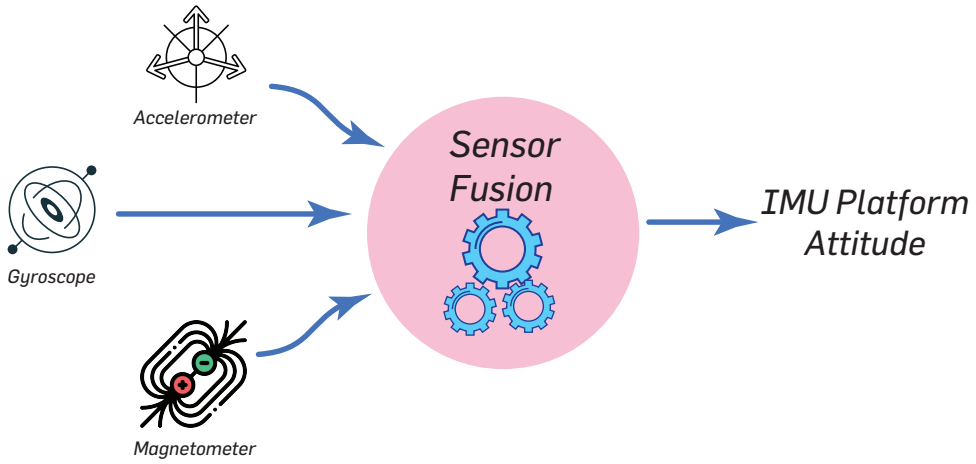
Research Focus



How to remove gravity from IMU accelerometer data ? How to implement it into FPGA?

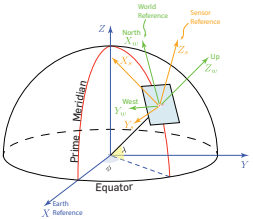
UNIVERSITY OF TWENTE.

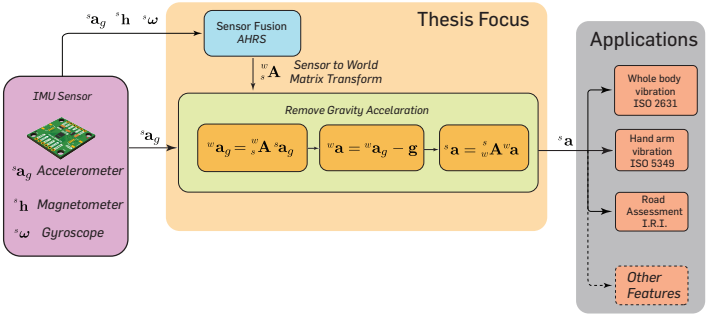
Sensor Fusion



UNIVERSITY OF TWENTE.

Application of Sensor Fusion results





UNIVERSITY OF TWENTE.

Vector Sensor to World Coordinate Frame Transform

Quaternion

Let the quaternion
$\underline{\mathbf{q}} = \underline{\mathbf{e}} = e_0 + e_1\vec{i} + e_2\vec{j} + e_3\vec{k}$ then:

$$ {}^{w}_{s}\underline{\mathbf{v}} = {}^{w}_{s}\underline{\mathbf{q}} \otimes {}^{s}\underline{\mathbf{v}} \otimes {}^{w}_{s}\underline{\mathbf{q}}^{*} $$

Rule for quaternion multiplication:

$$
\begin{aligned}
\underline{\mathbf{q}}_b \otimes \underline{\mathbf{q}}_a = {} & (q_{0a}q_{0b} - q_{1a}q_{1b} - q_{2a}q_{2b} - q_{3a}q_{3b}) + \\
& + (q_{0a}q_{1b} + q_{1a}q_{0b} + q_{2a}q_{3b} - q_{3a}q_{2b}) \cdot \vec{i} + \\
& + (q_{0a}q_{2b} - q_{1a}q_{3b} + q_{2a}q_{0a} + q_{3a}q_{1b}) \cdot \vec{j} + \\
& + (q_{0a}q_{3b} + q_{1a}q_{2b} - q_{2a}q_{1b} + q_{3a}q_{0b}) \cdot \vec{k}
\end{aligned}
$$

Matrix

$$ \mathbf{E} = \begin{bmatrix} -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix} $$

$$ \mathbf{G} = \begin{bmatrix} -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix} $$

$$ {}^{w}_{s}\mathbf{A} = \mathbf{E}\mathbf{G}^{\mathsf{T}} $$

$$ {}^{w}\mathbf{v} = {}^{w}_{s}\mathbf{A}\,{}^{s}\mathbf{v} $$

UNIVERSITY OF TWENTE.

Quaternions .vs. Matrices

	Quaternions	Matrices
Easiness of Manipulation	✓	✓
Computational Efficiency	✗	✓
Conciseness in Notation	✓	✗
Interpolation	✓	✗
Numerical Stability	✓	✗

Quaternions Properties

Quaternion components normalization property

$$\mathbf{q}^{\mathsf{T}}\mathbf{q} = e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$$

Quaternion consistency condition

$$E_m = \dot{\mathbf{q}}^{\mathsf{T}}\mathbf{q} = 0$$

Objective function to minimize

$$F = E_m^2$$

UNIVERSITY OF TWENTE.

Sensor Fusion of IMU Data

- Due to its importance and pervasive applications, Sensor Fusion is a commonly required task in IoT devices.
- For IMU sensor fusion, those of Mahony and Madgwick provided the theoretical ground work for many algorithms.
- None of them include in the computational flow the a consistency condition between quaternion and their time derivative.

Proposed Improvements of Madgwick Algorithm

- Translation of the Madgwick algorithm into matrix notation.
  Effect: Execution time reduced, easy operations scheduling.

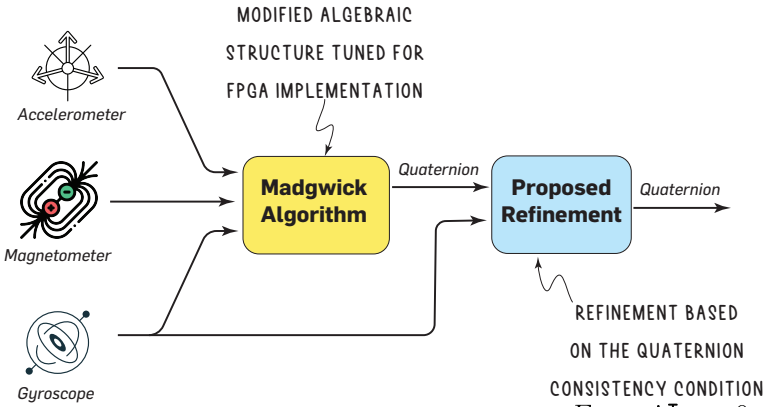
UNIVERSITY OF TWENTE.

Proposed Improvements of Madgwick Algorithm

- Translation of the Madgwick algorithm into matrix notation.
  Effect: Execution time reduced, easy operations scheduling.
- Simplification of the gradient algebraic structure.
  Effect: Execution time reduced, easy operations scheduling.

Proposed Improvements of Madgwick Algorithm

- Translation of the Madgwick algorithm into matrix notation.
  Effect: Execution time reduced, easy operations scheduling.
- Simplification of the gradient algebraic structure.
  Effect: Execution time reduced, easy operations scheduling.
- Addition of a quaternion consistency condition.
  Effect: Accuracy increase. A norm is introduced to evaluate IMU sensor fusion algorithms.


MODIFIED ALGEBRAIC
STRUCTURE TUNED FOR
FPGA IMPLEMENTATION
Accelerometer
Magnetometer
Gyroscope
Madgwick Algorithm
Quaternion
Proposed Refinement
Quaternion
REFINEMENT BASED
ON THE QUATERNION
CONSISTENCY CONDITION
$E_m = \dot{\mathbf{q}}^{\mathsf{T}}\mathbf{q} = 0$

The Modified Madgwick Algorithm

- **Data:** ${}^{s}\mathbf{a}_{i+1}$, ${}^{s}\mathbf{h}_{i+1}$, ${}^{s}\omega_{i+1}$, $\underline{\mathbf{q}}_{i}$
- **Constants:**
  ${}^{w}\mathbf{g} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^{\mathsf{T}}$, β, Δt, a_{th}, n
- **if** $||{}^{s}\mathbf{a}_{i+1}|| \leq a_{th}$ **then** $a =$TRUE **else** $a =$FALSE
- **Normalize:** ${}^{s}\mathbf{a}_{i+1} \leftarrow \dfrac{{}^{s}\mathbf{a}_{i+1}}{\left\|{}^{s}\mathbf{a}_{i+1}\right\|}$, ${}^{s}\mathbf{h}_{i+1} \leftarrow \dfrac{{}^{s}\mathbf{h}_{i+1}}{\left\|{}^{s}\mathbf{h}_{i+1}\right\|}$

- **Form** matrices $\mathbf{G}, \mathbf{E}$ and $\mathbf{A} = \mathbf{E}\mathbf{G}^{\mathsf{T}}$
- **Let**

$$^{w}\mathbf{h} = \mathbf{A}\,{}^{s}\mathbf{h}$$

- **Let**

$$^{w}\mathbf{h} = \begin{bmatrix} h_0 & h_1 & h_2 \end{bmatrix}$$

$$^{w}\mathbf{b} = \begin{bmatrix} \sqrt{h_0^2 + h_1^2} & 0 & h_2 \end{bmatrix}$$

- **Let**

$$^{w}\mathbf{b} = \begin{bmatrix} b_0 & 0 & b_1 \end{bmatrix}$$

- **if** $a == \text{TRUE}$ **then**

$$\mathbf{f}_a = \mathbf{A}\,{}^{w}\mathbf{g} - {}^{s}\mathbf{a}_{i+1}$$
$$\mathbf{f}_h = \mathbf{A}\,{}^{w}\mathbf{b} - {}^{s}\mathbf{h}_{i+1}$$
$$\mathbf{f} = \mathbf{f}_a + \mathbf{f}_h$$

else

$$\mathbf{f}_h = \mathbf{A}\,{}^{w}\mathbf{b} - {}^{s}\mathbf{h}_{i+1}$$
$$\mathbf{f} = \mathbf{f}_h$$

- **Let**

$$\mathbf{K} = \begin{bmatrix} -e_2 & e_3 & -e_0 & e_1 \\ e_1 & e_0 & e_3 & e_2 \\ 0 & -2e_1 & -2e_2 & 0 \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} 0 & 0 & -2e_2 & -2e_3 \\ -e_3 & e_2 & e_1 & -e_0 \\ e_2 & e_3 & e_0 & e_1 \end{bmatrix} \tag{1}$$

- **if** $a ==$ TRUE **then**

$$\mathbf{f}^\top \mathbf{J} = 2\left(\mathbf{f}_a^\top + b_1 \mathbf{f}_h^\top\right)\mathbf{K} + 2b_0 \mathbf{f}_a^\top \mathbf{L} \tag{2}$$

else

$$\mathbf{f}^\top \mathbf{J} = 2b_1 \mathbf{f}_h^\top \mathbf{K} + 2b_0 \mathbf{f}_a^\top \mathbf{L} \tag{3}$$

- **Compute**

$$\Delta \mathbf{q} = \frac{\mathbf{J}^\top \mathbf{f}}{||\mathbf{J}^\top \mathbf{f}||}$$

- **Compute:** $\dot{\mathbf{q}}_{\omega_{i+1}} = \frac{1}{2}\mathbf{G}^{\mathsf{T}}{}^{s}\omega_{i+1}$
- **Compute:** $\dot{\mathbf{q}}_{est_{i+1}} = \left(\dot{\mathbf{q}}_{\omega_{i+1}} - \beta\Delta\mathbf{q}\right)$
- **Compute:** $\mathbf{q}_{i+1} = \mathbf{q}_i + \dot{\mathbf{q}}_{est_{i+1}}\Delta t$
- **Normalize** $\mathbf{q}_{i+1}$

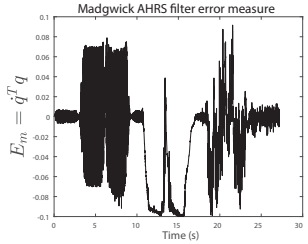
Proposed Refinement

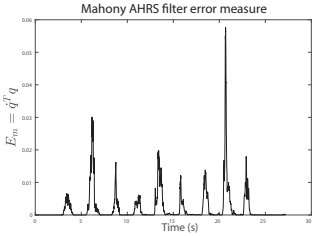
for $i = 1 : n$

- **Compute** E_m
- **Form G** with updated quaternion value $\mathbf{q}_{i+1}$
- **Update** $\dot{\mathbf{q}}_{\omega_{i+1}} = \frac{1}{2}\mathbf{G}^{\mathsf{T}}\, {}^{s}\boldsymbol{\omega}_{i+1}$
- **Compute** correction term $\Delta\mathbf{q}_{\mathrm{ref}} = -2\,\dot{\mathbf{q}}_{\omega_{i+1}} E_m^3$
- **Update** $\mathbf{q}_{i+1} \leftarrow \frac{\mathbf{q}_{i+1} - \Delta\mathbf{q}_{\mathrm{ref}}}{||\mathbf{q}_{i+1} - \Delta\mathbf{q}_{\mathrm{ref}}||}$

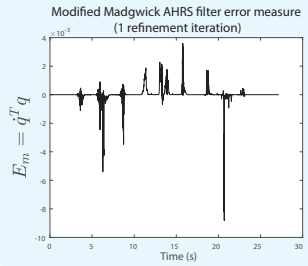
UNIVERSITY OF TWENTE.

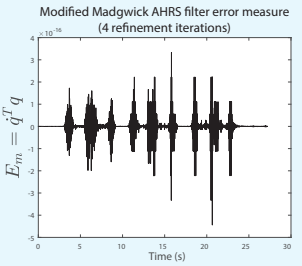
Results





This investigation





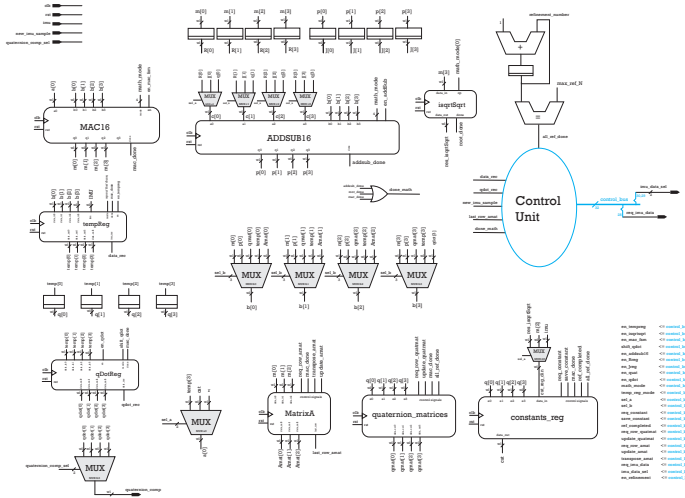
UNIVERSITY OF TWENTE.

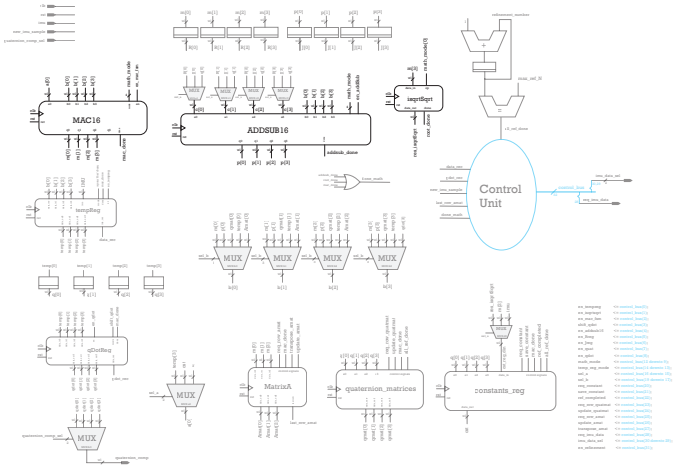
Comparison with other algorithms

	Madgwick	Mahony	Matlab IMU	Proposed Algorithm No Refinement Step	Proposed Algorithm 1 Refinement Step	Proposed Algorithm 4 Refinement Steps
Execution Time [s]	0.113	0.112	2.09	0.063	0.100	0.102
Max. Error $\lvert E_m \rvert$	0.0875	0.0576	2.2204e-16	0.0875	0.0116	3.3307e-16

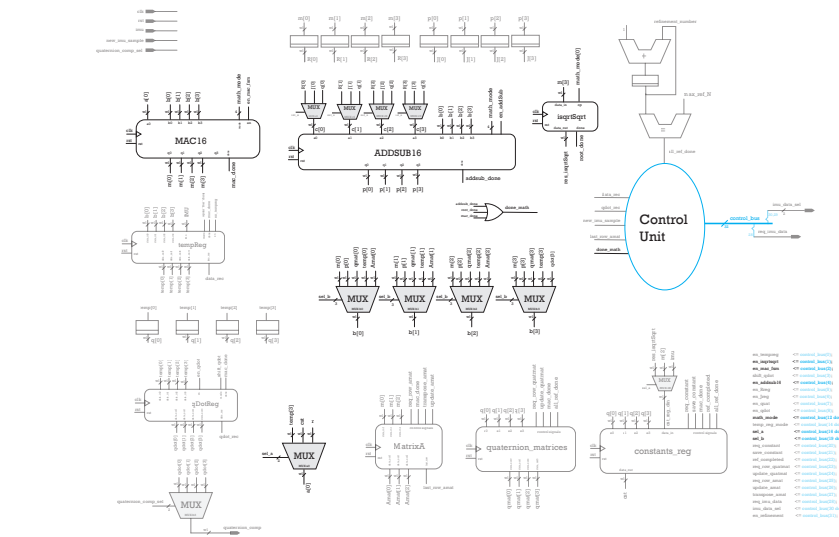
UNIVERSITY OF TWENTE.

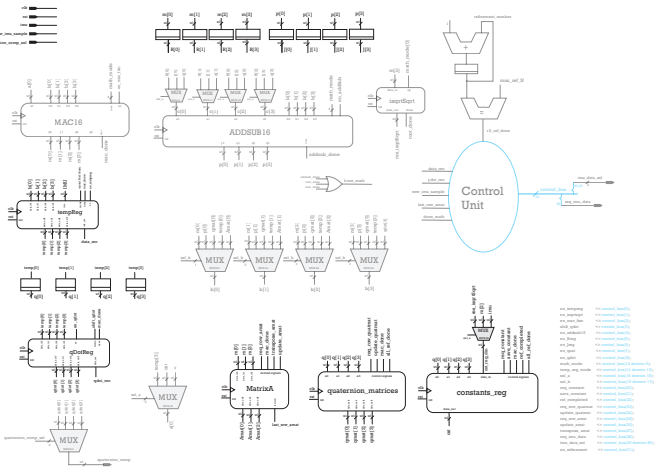
Proposed Hardware Architecture





- `MAC16` - vector scalar product, vector by scalar operations and matrix products.
- `ADDSUB16` - vectors additions.
- `isqrtSqrt` - computation of both ISQRT and SQRT.

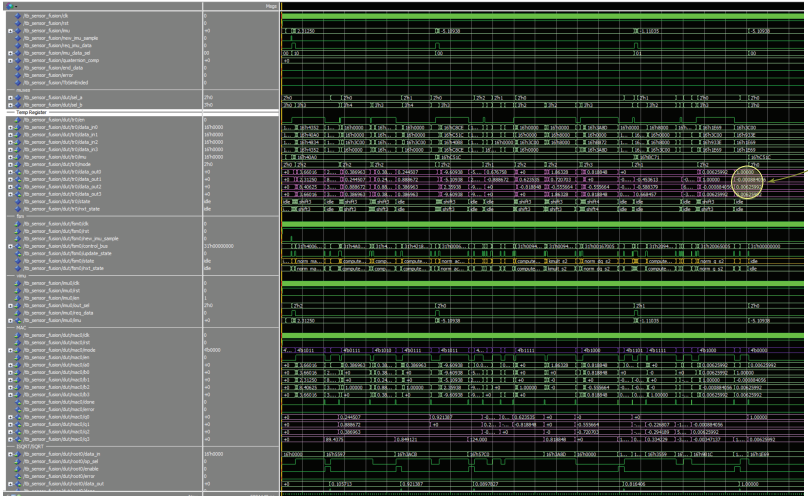

MAC16
ADDSUB16
MUX
tempReg
qDotReg
MatrixA
quaternion_matrices
constants_reg
Control
Unit
iagrtSqrt



- `tempReg` - register to hold temporary data.
- `MatrixA` - stores the content of matrix **A**.
- `quaternion_matrices` - stores quaternion dependent matrices (e.g. **E**,**L**,**K**).
- `constants_reg` - stores algorithm constants.

UNIVERSITY OF TWENTE.

Simulation Results



Quaternion elements results

Synthesis Results

Table: Resource Utilization

Logic Utilizations (in ALMs)	2392/29080 (8%)
Total Registers	2665
Total Block Memories	1865/4567040 (< 1%)
Total DSP Blocks	5/130 (3%)

Target FPGA: 5CEFA5F23C7 (Cyclone V Family)

UNIVERSITY OF TWENTE.

Timing Analysis

Table: Max. Frequency

Model	Max. Frequency [MHz]
Slow 1100mV 85C	62.41
Slow 1100mV 0C	62.63
Fast 1100mV 85C	114.80
Fast 1100mV 0C	128.35

A $\sqrt{x}$ and $1/\sqrt{x}$ Chebyshev polynomial approximation

Let us denote with $x \in \mathbb{R}^+$ a number on which to apply the operations of $\sqrt{x}$ or $\frac{1}{\sqrt{x}}$. We assume that this number is expressed according to `IEEE754` standard:

$$x = (-1)^s 2^{e_u} \cdot m \tag{4}$$

where $m \in [1\ ,\ 2)$ is the mantissa and e_u is the *unbiased* exponent ($e = e_u + \mathrm{bias}$). The following four cases can be distinguished:

- **isqrt even exponent**

$$\frac{1}{\sqrt{x}} = 2^{\frac{-e_U}{2}} \cdot \frac{1}{\sqrt{m}} \tag{5}$$

- **isqrt odd exponent**

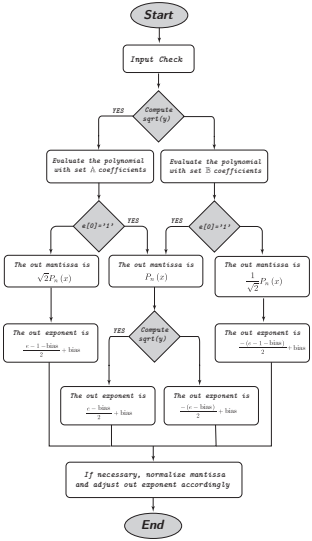
$$\frac{1}{\sqrt{x}} = 2^{\frac{-e_U+1}{2}} \cdot \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{m}} \tag{6}$$

- **sqrt even exponent**

$$\sqrt{x} = 2^{\frac{e_U}{2}} \cdot \sqrt{m} \tag{7}$$

- **sqrt odd exponent**

$$\sqrt{x} = 2^{\frac{e_U-1}{2}} \cdot \sqrt{2} \cdot \sqrt{m} \tag{8}$$


Start
Input Check
YES
Compute sqrt(y)
Evaluate the polynomial with set A coefficients
Evaluate the polynomial with set B coefficients
x[0]='1'
YES
YES
x[0]='1'
The out mantissa is $\sqrt{2}P_n(x)$
The out mantissa is $P_n(x)$
The out mantissa is $\frac{1}{\sqrt{2}}P_n(x)$
The out exponent is $\frac{e-1-\text{bias}}{2}+\text{bias}$
YES
Compute sqrt(y)
The out exponent is $\frac{-(e-1-\text{bias})}{2}+\text{bias}$
The out exponent is $\frac{e-\text{bias}}{2}+\text{bias}$
The out exponent is $\frac{-(e-\text{bias})}{2}+\text{bias}$
If necessary, normalise mantissa and adjust out exponent accordingly
End

The approximation of functions problem

- There is no scientific method of determining which approximating function $F_n(x)$ leads to the most efficient approximation of a function $f(x)$.
- Experience and intuition play an important role in choosing the approximating function

UNIVERSITY OF TWENTE.

Chebyshev .vs. Least-Square

	Least-Squares (Gauss)	Min-Max (Chebyshev)
Norm to be Minimized	$\sum \lvert f(x) - F_n(x) \rvert^2$	$\max_{x_0 \leq x \leq x_{n+1}} \lvert f(x) - F_n(x) \rvert$
Max. error Estimation	✗	✓
Points of max. error	✗	✓
Equioscillation of error func.	✗	✓

Chebyshev Optimality Criterion

Given a system of powers of x, $\left(1, x, x^2, x^3, \ldots, x^n\right)$, the Chebyshev min-max polynomial approximation of degree n of a continuous function $f(x)$, over an interval $[x_0, x_{n+1}]$, is the function

$$F_n(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n \tag{9}$$

such that $\varepsilon(x)$ is minimized, where

$$\varepsilon\left(x\right) = \max_{x_0 \leq x \leq x_{n+1}} |f(x) - F_n(x)| \tag{10}$$

It can be shown that $F_n(x)$ is the Chebyshev approximation of a differentiable function $f(x)$ on $[x_0, x_{n+1}]$ if and only if there exists a set of points

$$x_0 \leq x_1 \leq x_2 \leq \ldots \leq x_{n+1}$$

such that:

$$\varepsilon(x_i) = (-1)^i L \qquad (i = 0, 1, \ldots, n+1) \tag{11a}$$

$$\left.\frac{\mathrm{d}\varepsilon(x)}{\mathrm{d}x}\right|_{x=x_j} = 0 \qquad (j = 1, 2, \ldots, n) \tag{11b}$$

where L denotes the maximum value of ε in $[x_0, x_{n+1}]$.

The approximating polynomial for $1/\sqrt{x}$ and $\sqrt{x}$ is obtained by solving the following non linear system:

$$(-1)^{j+1} L + \sum_{i=0}^{n} p_i x_j^i - f(x) = 0 \quad (j = 0, 1, 2, \ldots, n+1) \tag{12a}$$

$$\sum_{i=1}^{n} i p_i x_k^{i-1} - \left.\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right|_{x=x_k} = 0 \quad (k = 1, 2, \ldots, n) \tag{12b}$$

whose solution yelds the unknowns L, $x_1, x_2, \ldots, x_n$, p_0, p_1, …, p_n. Therefore, not only the coefficients of the optimal approximating polynomial are computed, but also the value of maximum error L, as well as the abscissae where maxima and minima of $\varepsilon(x)$ occur.

UNIVERSITY OF TWENTE.

Approximation Accuracy Result

$1/\sqrt{x}$			$\sqrt{x}$		
n	N	MRE (this inv.)	n	N	MRE (this inv.)
2	1	3.8335e-03	2	1	7.6384e-04
2	2	7.1823e-04	2	2	1.4346e-04
2	4	1.1463e-04	2	4	2.2916e-05
2	8	1.6430e-05	2	8	3.2855e-06
2	16	2.2090e-06	2	16	4.4179e-07
2	32	2.8674e-07	2	32	5.7347e-08
2	64	3.6537e-08	2	64	7.3073e-09
3	1	5.7648e-04	3	1	8.2059e-05
3	2	6.3523e-05	3	2	9.0636e-06
3	4	5.5903e-06	3	4	7.9832e-07
3	8	4.2321e-07	3	8	6.0452e-08
3	16	2.9293e-08	3	16	4.1847e-09
3	32	1.9301e-09	3	32	2.7573e-10
4	1	8.9120e-05	4	1	9.8694e-06
4	2	5.7777e-06	4	2	6.4124e-07
4	4	2.8042e-07	4	4	3.1147e-08
4	8	1.1213e-08	4	8	1.2457e-09

ULP and relative error

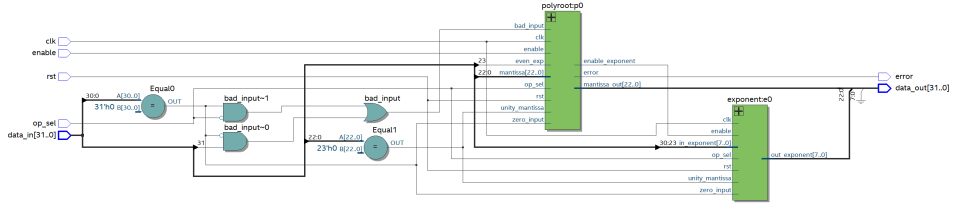
The relative error corresponding to $\frac{1}{2}$ ulp for a binary FP format with a precision of p bits is defined by

$$2^{-(p+1)} < \frac{1}{2}\,\mathrm{ulp} \leq 2^{-p} \tag{13}$$

- p for `IEEE 754` half precision is 10
- p for `IEEE 754` single precision is 23
- p for `IEEE 754` double precision is 51

UNIVERSITY OF TWENTE.

Proposed FPGA Architecture



Computation of output Mantissa

Let x be the input mantissa, the output mantissa is computed by evaluating the polynomial $P_2(x)$.
Four cases can be distinguished:

- **isqrt even exponent**

$$P_2(x) = (p_{20}x + p_{10})\,x + p_{00} \tag{14}$$

- **isqrt odd exponent**

$$P_2(x) = (p_{20}x + p_{10})\,\frac{x}{\sqrt{2}} + p_{01} \tag{15}$$

where $p_{01} = \dfrac{p_{00}}{\sqrt{2}}$

- **sqrt even exponent**

$$P_2(x) = (p_{22}x + p_{12})\,x + p_{02} \tag{16}$$

- **sqrt odd exponent**

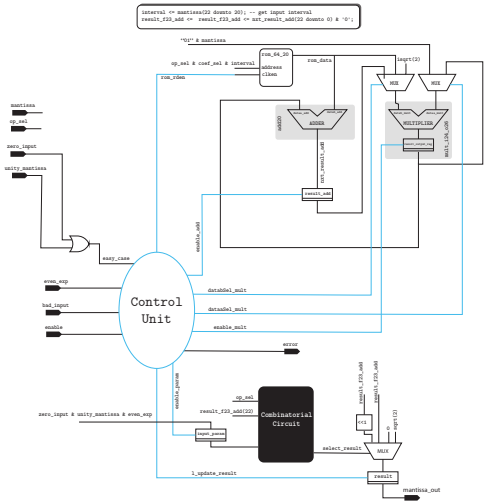
$$P_2(x) = 2\left[(p_{22}x + p_{12})\,\frac{x}{\sqrt{2}} + p_{03}\right] \tag{17}$$

where $p_{03} = \dfrac{p_{02}}{\sqrt{2}}$

Approximating Polynomial Evaluation

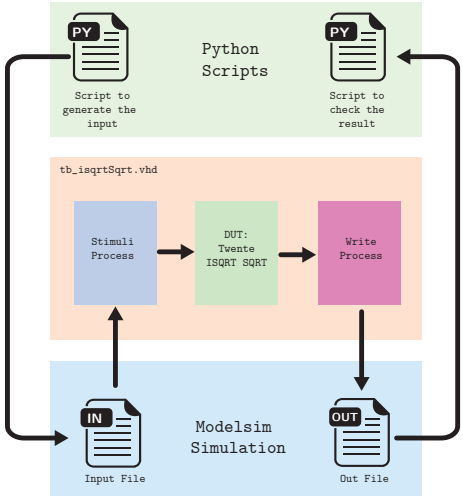
- **Hardware Resources**:
  - $1\times$ Adder
  - $1\times$ Multiplier
  - $1\times$ ROM
  - Registers to hold $\frac{1}{\sqrt{2}}$ constant, multiplication/addition results and the result mantissa.
- **Operation Scheduling** $P_2(x) = (p_{22}x + p_{12})\,kx + p_{02}$

	0	1	2	3
Multiplier	$m \leftarrow p_2 x$	$m \leftarrow x\frac{1}{\sqrt{2}}$ or $m \leftarrow x$	$m \leftarrow ma$	
Adder		$a \leftarrow m + p_1$		$r \leftarrow m + p_0$


interval <= mantissa(22 downto 20); -- get input interval
result_f23_add <= result_f23_add <= nxt_result_add(32 downto 0) & '0';
"01" & mantissa
rom_64_32
address
clken
rom_data
op_sel & coef_sel & interval
rom_rden
input(2)
MUX
MUX
mantissa
op_sel
zero_input
unity_mantissa
ADDER
dataa_sum
datab_sum
result_sum
nxt_result_add
MULTIPLIER
dataa_mult
datab_mult
result_mult
MULT_SUM_ADD
easy_case
enable_add
even_exp
bad_input
enable
Control
Unit
databdel_mult
datadel_mult
enable_mult
error
enable_param
op_sel
result_f23_add(22)
zero_input & unity_mantissa & even_exp
input_param
Combinatorial
Circuit
select_result
nxt_result_add
result_f23_add
<<1
0
input(2)
MUX
result
l_update_result
mantissa_out

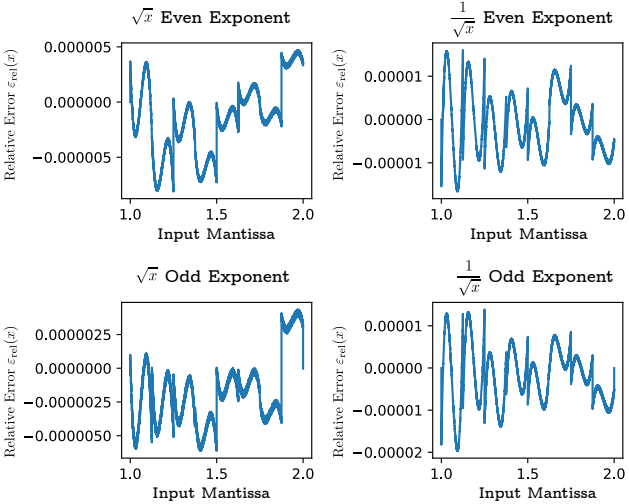
UNIVERSITY OF TWENTE.

Simulation Strategy



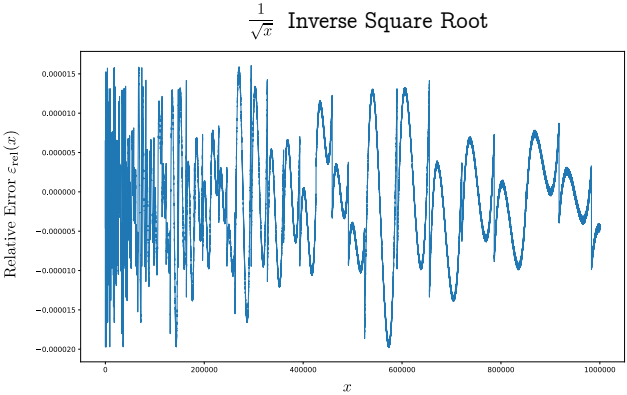
UNIVERSITY OF TWENTE.

Simulation Results: Mantissa



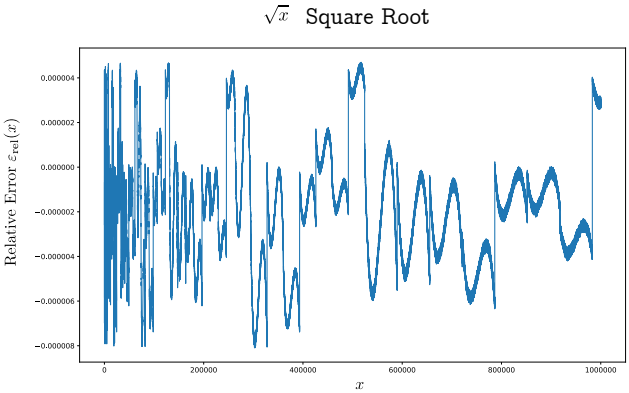
UNIVERSITY OF TWENTE.

Simulation Results: Inverse Square Root



UNIVERSITY OF TWENTE.

Simulation Results: Square Root



UNIVERSITY OF TWENTE.

Synthesis Results

	Proposed	ALTFP_INV_SQRT	ALTFP_SQRT
Logic Utilizations (in ALMs)	66/9430 (< 1%)	278/9430 (3%)	245/9430 (3%)
Total Registers	60	925	542
Total Block Memories	1280/1802240 (<1%)	565/1802240 (<1%)	143/1802240
Total DSP Blocks	1/25 (4%)	6/25 (24%)	0/25 (0%)
Latency in clk cycles	7	26	16

Target FPGA: 5CEFA2F23I7 (Cyclone V Family)

UNIVERSITY OF TWENTE.

Timing Analysis

Table: Max. Frequency

Model	Max. Frequency [MHz]
Slow 1100mV 100C	127.28
Slow 1100mV -40C	123.87
Fast 1100mV 100C	230.47
Fast 1100mV -40C	260.28

UNIVERSITY OF TWENTE.

Conclusions

This work tackled the problem of estimating the spatial orientation of a single 9-DoF IMU platform. The following goals have been achieved:

- Functional outlines of a novel device for bikers comfort vibration monitoring and road condition assessment.

UNIVERSITY OF TWENTE.

Conclusions

This work tackled the problem of estimating the spatial orientation of a single 9-DoF IMU platform. The following goals have been achieved:

- Functional outlines of a novel device for bikers comfort vibration monitoring and road condition assessment.
- Improvement of the Madgwick sensor fusion algorithm both in speed and accuracy.
- A custom architecture to execute the proposed sensor fusion algorithm was implemented.

UNIVERSITY OF TWENTE.

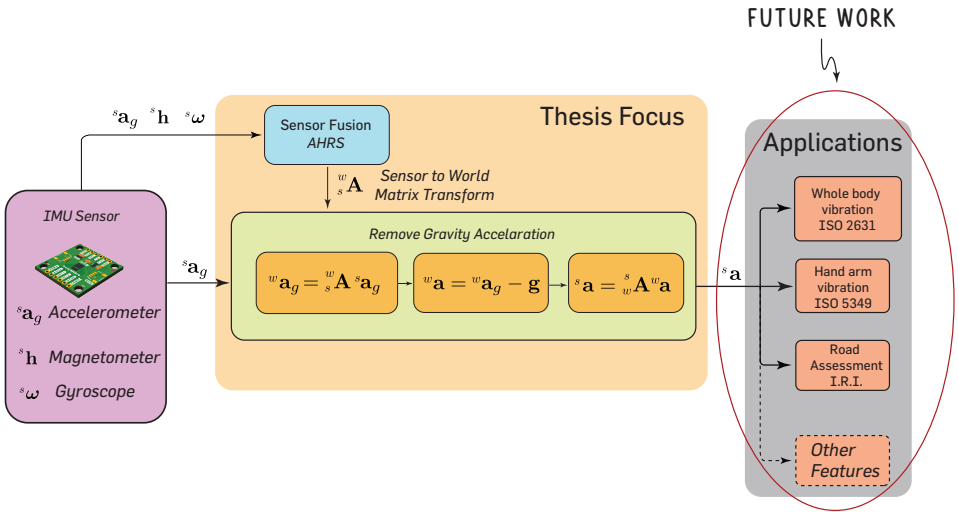
Conclusions

This work tackled the problem of estimating the spatial orientation of a single 9-DoF IMU platform. The following goals have been achieved:

- Functional outlines of a novel device for bikers comfort vibration monitoring and road condition assessment.
- Improvement of the Madgwick sensor fusion algorithm both in speed and accuracy.
- A custom architecture to execute the proposed sensor fusion algorithm was implemented.
- Development of an original method and architecture to compute, with shared resources, both $\frac{1}{\sqrt{x}}$ and $\sqrt{x}$ with a polynomial approximation.

UNIVERSITY OF TWENTE.

Future Work



UNIVERSITY
OF TWENTE.

Thank you for your attention.